

AD-A048 769

COMPUTER SCIENCES CORP SAN DIEGO CALIF
SIMULATION CODE FOR UNDERWATER TELEVISION SYSTEMS (SCOUTS). (U)
SEP 77 A GORDON, T SELDEN

F/G 17/2

N00123-74-C-0312

UNCLASSIFIED

NOSC-TR-146

NL

1 OF 1
AD
A048769



NOSC TR 146 AD A 048769

12
NOSC

NOSC TR 146

Technical Report 146

SIMULATION CODE FOR UNDERWATER TELEVISION SYSTEMS (SCOUTS)

A Gordon
Naval Ocean Systems Center
T Selden
Computer Sciences Corporation
1 September 1977

Interim Report: June 1976 — May 1977
Prepared For
Naval Sea Systems Command



Approved for public release; distribution unlimited.

NAVAL OCEAN SYSTEMS CENTER
SAN DIEGO, CALIFORNIA 92152

DDC FILE COPY



NAVAL OCEAN SYSTEMS CENTER, SAN DIEGO, CA 92152

AN ACTIVITY OF THE NAVAL MATERIAL COMMAND

RR GAVAZZI, CAPT USN

Commander

HL BLOOD

Technical Director

ADMINISTRATIVE INFORMATION

Work on this report was performed from 1 June 1976 to 1 May 1977. The work was funded as part of the Deep Ocean Technology Project by the Naval Sea Systems Command under Program Element 63713N. This report was reviewed for technical accuracy by Dr. Peter McCardell of the Naval Ocean Systems Center, San Diego. Work by CSC was performed under Contract N-00123-74-C-0312.

Released by
IP Lemaire, Head
Advanced Systems Division

Under authority of
HR Talkington, Head
Ocean Technology Department

(18) NOSE

(19) TR-146

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NOSC Technical Report 146 (TR 146)	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) SIMULATION CODE FOR UNDERWATER TELEVISION SYSTEMS (SCOUTS).	5. TYPE OF REPORT & PERIOD COVERED (9) Research and Development 1 Jun 76 - 1 May 77, rept.	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Alan Gordon Naval Ocean Systems Center Thomas Selden Computer Sciences Corporation	8. CONTINUING OR GRANT NUMBER(s) (15) N00123-74-C-0312	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Ocean Systems Center San Diego, CA 92152	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Element 63713N	
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Sea Systems Command Washington, D.C.	12. REPORT DATE (11) Sept 77	13. NUMBER OF PAGES (12) 44p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) UNCLASSIFIED	15a. DECLASSIFICATION DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) underwater television computerized simulation FORTRAN IV		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Based on the system performance equations developed in the Naval Undersea Center's NUC TP 303, Handbook of Underwater Imaging System Design, a computer code was written to predict the performance of underwater television systems. This code, known as SCOUTS (for "Simulation Code for Underwater Television Systems") is written in FORTRAN IV and is suitable for demand or batch runs. SCOUTS allows the user to select environmental, geometric and hardware parameters. It can be used for system design as well		

DDC
JAN 19 1978
F

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

390612

Inuc

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. Continued.

as analysis. SCOUTS includes a parametric mode which allows automatic variation of multiple parameters. A complete source listing is included as an Appendix.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

SUMMARY

PROBLEM

Using the system performance equations developed in NUC TP 303, Handbook of Underwater Imaging System Design, develop a computer code for predicting the performance of underwater television systems.

RESULTS

A code called SCOUTS (for "Simulation Code for Underwater Television Systems") and suitable for batch or demand runs has been written in FORTRAN IV. SCOUTS allows the user to select environmental, geometric, and hardware parameters. It can be used for system design as well as analysis. SCOUTS includes a parametric mode that allows automatic variation of multiple parameters.

BY	
DISTRIBUTION/AVAILABILITY CODES	
SPECIAL	
Dis	
A	

CONTENTS

1.0 INTRODUCTION	5
1.1 Need for Performance Modeling of Underwater Television	5
1.2 The Handbook of Underwater Imaging System Design	5
1.3 Evolution of SCOUTS	6
2.0 OVERVIEW OF SCOUTS	7
2.1 System Geometry	7
2.2 User Inputs	8
2.3 Types of Calculations Available	10
2.4 Output	12
3.0 BASIS OF SCOUTS' CALCULATIONS	15
3.1 Water Characteristics	15
3.2 Hardware Characteristics	15
3.3 Performance Calculations	16
4.0 PROGRAM STRUCTURE	17
4.1 Subroutine Organization	17
4.2 Subroutine and Variable Definitions	18
5.0 PROGRAM OPERATION	21
5.1 Preliminary Requirements	21
5.2 Batch Processing	22
5.3 Demand Terminal Use	23
REFERENCES	24
APPENDIX	25

LIST OF FIGURES

1. SCOUTS system geometry.	7
2. Sample input for demand run.	9
3. Intermediate summary.	13
4. Final summary.	14
5. Program heirarchy.	17
6. Creation of absolute element.	21
7. Sample deck setup for batch execution of VSEARCH.	23

LIST OF TABLES

1. User inputs.	8
2. Variable names and definitions.	19
3. SCOUTS subroutines.	20
4. Batch mnemonics.	22

1.0 INTRODUCTION

This report documents the Naval Ocean Systems Center's (NOSC) Simulation Code for Underwater Television Systems (SCOUTS). SCOUTS is a real time, interactive (when used in the DEMAND mode) computer code which is useful in the design, analysis and performance prediction of a large class of underwater television systems. A complete printout of SCOUTS in the FORTRAN IV language is included in the Appendix. For those with immediate applications, the information in Sections 2 and 5 is sufficient to describe the operation and use of SCOUTS. Sections 3 and 4 give the theoretical basis for the calculations and internal program structure, respectively. The remainder of this Section concerns itself with the background and rationale of SCOUTS' development.

1.1 NEED FOR PERFORMANCE MODELING OF UNDERWATER TELEVISION

During the past decade and a half man has rapidly extended his presence beneath the surface of the ocean. Manned submersibles, tethered vehicles, moored drill rigs and even underwater habitats reflect his desire to understand and utilize the unique world beneath the ocean's surface. Underwater photography has been used throughout this era to document subsurface missions. This time period has also seen the development of underwater television to the point that it is equal in importance to photography as an undersea visual sensor.

Modern underwater television systems use lenses, light sources and cameras specifically designed for this environment. These components are incorporated into system concepts which attempt to minimize backscatter while maintaining an acceptable signal-to-noise ratio. As a result of the progress made during the last 15 years, the designer of underwater television systems now can choose from a wide variety of components and techniques in tailoring the system to his particular requirements.

One rather disturbing effect of the availability of advanced components and techniques was that predicting the performance of the resulting system became much more difficult. Simple single scattering models gave erroneous results at the longer ranges corresponding to advanced imaging systems. More exact models were available, but they were primarily theoretical and not generally suitable for engineering use. Thus, by the early 1970s, a situation arose where hardware was available which appeared to promise improved system performance, but the methodology for comparing the available hardware and accurately predicting performance was lacking.

1.2 THE HANDBOOK OF UNDERWATER IMAGING SYSTEM DESIGN

Under the sponsorship of the Navy's Deep Ocean Technology Program, the Naval Ocean Systems Center undertook the task of providing a definitive handbook for the designer of underwater television systems. This effort produced the Handbook of Underwater Imaging System Design (HUISD) (Reference 1). HUISD provided a detailed discussion of optical water parameters, propagation of light underwater, characteristics of system components and

advantages and drawbacks of system concepts then in use or proposed. More importantly, HUISD gave a detailed design procedure for obtaining measures of system performance based on the above factors. This design procedure, validated by comparative system tests performed by NOSC at Morris Dam in 1970, was presented both as a set of nomograms and a step-by-step series of equations. These equations, which were more accurate and flexible than the nomograms, required lengthy hand calculations and were tedious to implement. Consequently, these system equations were not applied as widely as had been hoped, though their predictions proved accurate in those instances where the hand calculations were performed.

1.3 EVOLUTION OF SCOUTS

About a year after the publication of HUISD, NOSC began work on the computer modeling portions of the Advanced Unmanned Search System (AUSS) program. Part of the AUSS model required the simulation (i.e., performance prediction) of underwater television systems. Since the HUISD equations were already developed and validated, these equations were used as the basis for the AUSS computer code. Only the HUISD performance analysis of conventional systems (i.e., systems using non-laser sources and raster-scanned receivers) was coded, since these systems were the ones most commonly in use. The first AUSS TV program was coded in BASIC, but in order to achieve faster operating times the program was rewritten in the FORTRAN language.

Experience with the AUSS TV program quickly showed that the HUISD system performance equations, coupled with real time, interactive demand terminal computing, provided a powerful tool for TV system design and analysis. Using the program the operator was immediately furnished with the performance of the selected configuration as well as indications as to what (e.g., source power, backscatter) were the limiting factors. By successively adjusting the TV system's configuration the operator could optimize it for his particular arrangement. This procedure was successfully used in designing and validating systems used in search, small-object detection, obstacle avoidance, etc.

One of the problems with the AUSS TV program was that it is a subprogram integral to the much larger AUSS model. In order to use just the TV portion, much information extraneous to the operation of the TV subsystem has to be keyed in. Because there was a significant number of AUSS users who were interested primarily in the TV subprogram, it was decided to develop SCOUTS as an independent computer code for the simulation of underwater television.

SCOUTS differs in a few important ways from the AUSS TV subprogram. SCOUTS is coded in FORTRAN IV and should be compatible as written with most medium- and large-size computers. A parametric capability allows several variables to be automatically varied according to limits defined by the user. Finally, although SCOUTS is primarily demand terminal-oriented, a batch option is included for computer facilities lacking real time capabilities.

2.0 OVERVIEW OF SCOUTS

This section will be concerned with defining the problem which SCOUTS solves, the inputs required and the outputs provided. Coupled with the detailed operating instructions in Section 5, this section provides sufficient information for successful operation of the program.

2.1 SYSTEM GEOMETRY

The geometry assumed by SCOUTS for its calculations is shown in Figure 1. The source and receiver are spatially separated by the source-receiver separation, d . Without loss of generality the source is assumed to be to the receiver's right when looking out towards the target. Their optical axes are coplanar and each is "canted in" by an angle δ so that a perpendicular dropped from the axes' intersection point will bisect the line joining the source and receiver. The receiver's (i.e., TV's) horizontal scan direction is parallel to the line joining source and receiver.

The target's center is in the plane containing the source and receiver's optical axes (i.e., the plane of the paper). The long dimension of the object is in the plane which is orthogonal to the plane of the paper and parallel to the line joining source and receiver. The

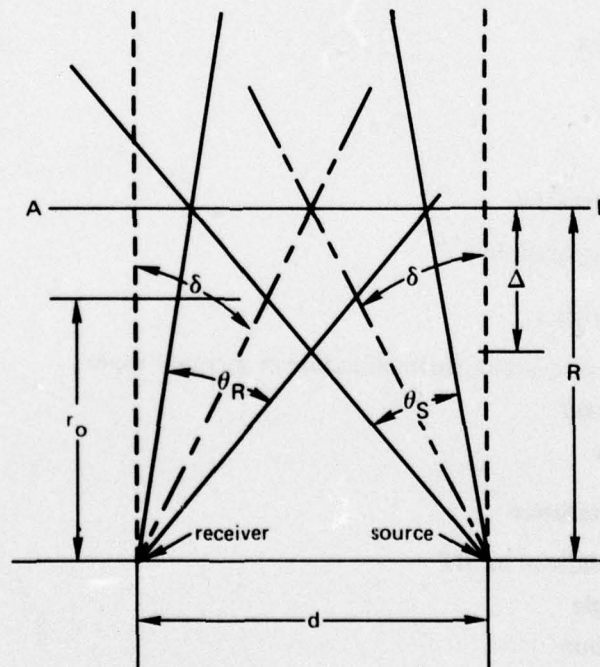


Figure 1. SCOUTS system geometry.

point of the object most distant from the source-receiver plane (i.e. the plane containing the source and receiver and perpendicular to the plane containing their optical axes) is assumed to be situated a distance R from it. The height of the object, H , is the extent of the object perpendicular to the source-receiver plane.

The distance r_0 in Figure 1 is the distance past which backscatter contributes to the image. The depth of field, Δ , is the maximum distance from the target plane at which there is a "common volume" where source and receiver beams intersect any portion of the target whose height exceeds Δ will not be imaged.

2.2 USER INPUTS

Table 1 contains the inputs necessary to define the imaging problem for SCOUTS. All these variables must be entered on the initial calculation of any run. A rewrite capability (see Section 2.5) is included so that this information does not have to be entered each time it is desired to change only some of the variables. An example of the initial input for a demand run is shown in Figure 2.

Table 1. User inputs.

Environment
1. Water type: coastal or deep
2. Peak-to-trough wave height
3. Peak-to-trough bottom roughness
Target Dimensions
4. Target length
5. Target height
Platform Characteristics
6. Source-receiver separation
Source Characteristics
7. Source type: incandescent, thallium iodide or mercury vapor
8. Source input power
9. Source full angle
Receiver Characteristics
10. Receiver type: vidicon or SIT
11. Receiver full angle
12. Optics transmission
13. F-number

CONVENTIONAL TV SYSTEM

ENVIRONMENT
COASTAL OR DEEP, 1 OR 2:?
>1
WAVE HEIGHT-PEAK TO TROUGH, FT:?
>2.
BOTTOM ROUGHNESS-PEAK TO TROUGH, FT:?
>.5
TARGET DIMENSIONS
LENGTH OF TARGET, FT:?
>10.
HEIGHT OF TARGET, FT:?
>.5
PLATFORM CHARACTERISTICS
SOURCE RECEIVER SEPARATION, FT:?
>1.
SOURCE CHARACTERISTICS
INCANDESCENT, THALLIUM IODIDE, OR MERCURY VAPOR; 1, 2 OR 3:?
>2
SOURCE POWER, WATTS:?
>200.
FULL SOURCE ANGLE, DEG:?
>40.
RECEIVER CHARACTERISTICS
VIDICON OR SIT, 1 OR 2:?
>1
FULL RECEIVER ANGLE, DEG:?
>40.
TRANSMISSION OF OPTICS:?
>.9
F-NUMBER:?
>1.5

Figure 2. Sample input for demand run.

Table 2 is for the most part self-explanatory; however, a few comments are in order. SCOUTS was originally written for an object lying on the bottom viewed from a TV camera which was responding to ocean heave. To ensure an adequate depth of field, Δ , SCOUTS chooses Δ so that

$$\Delta = H + 0.5 * (h_1 + h_2) \quad (1)$$

where h_1 and h_2 are the wave height and bottom roughness respectively. For situations where the target is not on the bottom, h_1 and h_2 should be set to zero.

SCOUTS incorporates information from HUISD to provide the appropriate characteristics for the water, lamp and receiver types. These characteristics will be discussed in detail in Section 3. However, in order for the user to supply meaningful inputs, it is necessary to discuss some of these characteristics here.

The two water types, coastal and deep, have optical characteristics identical with those plotted in Figures 6.2(B) and 6.2(C) of HUISD. The coastal water has a minimum attenuation, α , of 0.253/m ($1/\alpha = 3.95$ m) at $\lambda = 540$ nm. For the deep water, the corresponding numbers are $\alpha = 0.049$ /m ($1/\alpha = 20.4$ m) at 475 nm. The scattering coefficient, s , is 0.238/m for the coastal water and 0.030/m for deep water.

For sources with conical beam patterns, the required "full source angle," θ_s , is just the apex angle of the in-water beam pattern. For a nonconical source, the "equivalent conical beam pattern" (HUISD, p. 6-11, Eq. 6.10) must be determined and its apex angle used as input. For "source power," SCOUTS requires the electrical power input to the lamp, then computes the light power output.

The full receiver angle, θ_R , is the full in-water horizontal field of view of the TV camera. SCOUTS assumes the standard 4-by-3 aspect ratio for the receiver's field of view.

For transmission of optics SCOUTS requires the overall decimal efficiency of the source and receiver. This is the combined efficiency of all the optical elements following the source and up to and including the camera lens. For example, if the collection efficiency of the light source and the transmissivity of the camera lens were each 0.9, the required optical transmissivity would be 0.81.

The f-number, $f/$, is defined for small angles by

$$f/ = \frac{f}{D} \quad (2)$$

where f is the focal length and D the receiver aperture diameter. The $f/$ has a minimum value of 0.5. Most camera lenses will indicate the $f/$ corresponding to each aperture stop.

2.3 TYPES OF CALCULATIONS AVAILABLE

2.3.1 Demand and Batch Runs

SCOUTS can be run either as a batch or demand run. It is important to remember that SCOUTS was designed for real time, interactive computing, i.e. for demand runs. Design problems are most efficiently handled in the demand mode, since the user will be able to refine his design based on immediate feedback of the performance of his previous choice. Another advantage is that after a period of running design problems via demand runs, the user will develop his intuition for good underwater systems design.

The batch run capability was included primarily for computer systems not having real time facilities. However, the batch mode can be used to advantage in some situations. Some computer facilities charge substantially less for batch runs than for real time operation. Although SCOUTS' costs are low because less than 1.5 sec combined CAU (central arithmetic unit) and ER (executive request) time per calculation is required on the NOSC Univac 1110, the cost savings might be significant if many calculations are required. Batch runs necessarily provide a hard-copy output; this output may be absent or of poor quality on demand terminals. In very large runs using the parametric mode (see Section 2.3.2) where many variables are to be parametrized in a noninteractive fashion, using the batch mode allows the program to run without the operator being present.

2.3.2 Nonparametric and Parametric Modes

Nonparametric and parametric modes are available for both demand and batch runs. In the nonparametric mode, each input variable (see Table 1) takes on only one value and a calculation is immediately performed. In the parametric mode, the user selects one or more variables which he parametrizes by assigning each a beginning value, incremental step and final value. SCOUTS then performs as many calculations as necessary to evaluate the system for all the selected values of the input variables, and the results are output.

The parametric mode is thus similar to a succession of nonparametric calculations. The user selects the mode appropriate for his objectives. In optimizing a system, a succession of nonparametric calculations will allow the user to select new input values based on the results of past calculations. The performance of a selected system for various values of input data can be documented more quickly in the parametric mode.

2.3.3 Calculational Modes

There are three calculational modes available to the user: single range, maximum range or maximum swath width. The calculational mode is chosen in demand runs by the appropriate response to a SCOUTS question and in batch runs by inclusion of the appropriate mnemonic (see Table 4). In the single-range mode the system performance is evaluated at the range supplied by the user. To understand the other two modes we must first define what is meant by usable swath width.

The usable (or actual) swath width at range R is measured along that line in the source-receiver-target plane which is parallel to the line joining source and receiver and a distance R from it (the line AB in Figure 1). A segment of this line will have the following two properties: (1) a target located on this segment will have an image contrast greater than or equal to 0.07 and (2) the target's image will be large enough to span eight or more resolved TV lines. The length of this segment is taken to be the actual swath width. The actual swath width thus represents the lateral dimension in which successful imagery is possible.

As a function of range the swath width increases almost linearly with range, reaches a maximum and then drops off rapidly as the range is increased further. The maximum range calculation evaluates the system at that range where the swath width, after having passed its maximum, drops to zero. Similarly, the maximum swath width calculation evaluates the system at the range which yields the greatest swath width.

In the nonparametric mode, SCOUTS can evaluate the system in any of the three calculational modes. In the parametric mode, the user can have the system evaluated at either the maximum range or maximum swath width by indicating his choice and setting the range equal to zero. If the range has some other value the system will be evaluated at the selected range; however, SCOUTS will, at the user's option, calculate the value of the maximum range, maximum swath width, or both. Since SCOUTS performs these maximizing calculations by performing a series of single-range calculations, running time and costs can be minimized by selecting maximizing calculations only when necessary.

2.4 OUTPUT

2.4.1 Minimum Range

If the target is located too close to the source-receiver plane TV performance can be degraded. At distances which are too close, the resultant depth of field becomes less than the minimum value required by Eq. 1. Additionally, at distances sufficiently short the source will be pointing directly towards the receiver and blind it. SCOUTS therefore calculates a minimum range prior to evaluating the user's system. In demand runs this information is presented to the user in time for the user to choose a range greater than the minimum. SCOUTS then checks that the input range is indeed greater than the minimum. If not, the calculation will not be performed.

One of the disadvantages of batch runs is that the user does not receive an indication of the allowable minimum range until the run is over. Thus, there is an uncertainty as to the smallest range that will execute successfully. Since SCOUTS sets the minimum range no smaller than the source-receiver separation, batch input ranges should be larger than this value. This will not guarantee a successful run but is the best the user can do with no other *a priori* information.

2.4.2 Summary Output

The results of a SCOUTS calculation (or calculations in the parametric mode) represent the performance evaluation of the specified TV system. This performance evaluation is provided by SCOUTS in the form of two output summaries, the intermediate summary (Figure 3) and the final summary (Figure 4). Intermediate summaries are provided at the end of each demand nonparametric run, while final summaries are output in all modes. Both summaries contain essentially the same information in different formats. The meaning of each of the outputs will now be presented in the order in which they appear on the final summary.

The first thirteen columnar entries on Figure 3, from WATER TYPE to F-NUMBER, reproduce the user-selected input so that a record of the evaluated system is available along with the output. RANGE, FT and RANGE, AL give the range at which the calculation was performed in feet and attenuation lengths, respectively. All the performance measures from AVAILABLE LINES AT CTR to the second to the last entry refer to these measures evaluated at this range. MINIMUM, FT is the minimum allowable range (See Section 2.4.1) in feet. If the maximum range or maximum swath width options are chosen, the range at which each of these occurs is printed out in feet after MAXIMUM, FT and BEST, FT, respectively.

The number of resolved TV lines spanning the object length is given by AVAILABLE LINES AT CTR for the object located at the center of the receiver's field of view and by AT EDGE for the object located at the right-hand edge of the receiver's field of view. Because of the interpolation subroutine used, the number of lines may be in excess of 525, which is taken to be the display limit. It has been shown that eight lines represent the threshold of object recognition (Reference 3).

SCOUTS calculates the image contrast for an object having an inherent contrast of 50 percent and a highlight reflectivity of 75 percent. CONTRAST AT CENTER and AT EDGE are respectively the image contrast for a target located at the center and right-hand edge of

CONVENTIONAL TV SYSTEM

WATER TYPE:	COASTAL	SOURCE:	THAL. IOD.
WAVE HEIGHT, FT:	2.00000	SOURCE POWER:	200.00000
BOTTOM, FT:	.50000	SOURCE BEAM:	40.00000
LENGTH, FT:	10.00000	RECEIVER:	VIDICON
HEIGHT, FT:	.50000	RECEIVER BEAM:	40.00000
TAU:	.90000		
S.R. SEP.:	1.00000	F/:	1.50000
RANGE, FT:	24.40228	RANGE, AL:	1.87424
AVAILABLE LINES AT CTR.:	393.90030	AT EDGE:	393.90030
CONTRAST AT CENTER:	.16397	AT EDGE:	.07523
AVAILABLE L H S SW WIDTH = 8.81963 FT		ACTUAL = 8.22299 FT	
LIMITED BY:R. ANG			
AVAILABLE R H S SW WIDTH = 8.95223 FT		ACTUAL = 8.22299 FT	
LIMITED BY:S. ANG			
AVAILABLE SW WIDTH = 17.77186 FT		ACTUAL = 16.44598 FT	

Figure 3. Intermediate summary.

the receiver's field of view. Since backscatter is most severe towards the right edge of the field of view (remember that SCOUTS assumes the light is to the right of the receiver) the contrast at this point (as well as the number of lines available) will always be less than that in the center.

The next eight entries on the final summary give information on the swath width to be expected. AV L.H.S. (R.H.S.) SW WIDTH, FT is the available left-hand (right-hand) swath width. This is the segment, in feet, of the swath width line (See Section 2.3.3) to the left (right) of the center of the receiver's field of view which is illuminated by both source and receiver beams. ACTUAL, FT is the portion of the available left-hand (right-hand) swath width, in feet, where the target may be successfully imaged. Again, the three criteria used by SCOUTS for successful imaging are: (1) adequate depth of field as defined by Eq. (1), (2) at least eight resolved TV lines across the target's longest dimension and (3) an image contrast of at least 7 percent.

LIMITED BY gives a mnemonic which indicates what is limiting the extent of the R.H.S. (L.H.S.) actual swath width. The mnemonics S. ANG. and R. ANG refer to source angle and receiver angle respectively and indicate that the corresponding beam's geometry is limiting the available swath width. BSKTR indicates that the available swath exceeding 7-percent contrast is the limiting factor. Similarly, PWR signifies that the available swath width was limited by the requirement for eight TV lines/object length. DISPLY is printed when the number of lines/raster height required to place eight TV lines across the image's longest dimension exceeds 525.

AV SWATH WIDTH, FT is the arithmetic sum of the available left- and right-hand swath width and indicates the total available swath width for the given geometry and range. ACTUAL, FT is the corresponding sum of the actual swath widths, which is the linear

WATER TYPE:	COASTAL	COASTAL	COASTAL	COASTAL
WAVE HEIGHT, FT:	2.0000	2.0000	2.0000	2.0000
BOTTOM ROUGHNESS, FT:	.5000	.5000	.5000	.5000
TARGET LENGTH, FT:	10.0000	10.0000	10.0000	10.0000
TARGET HEIGHT, FT:	.5000	.5000	.5000	.5000
S.R.SEP., FT:	1.0000	1.0000	2.0000	3.0000
SOURCE:	THAL. IOD.	THAL. IOD.	THAL. IOD.	THAL. IOD.
SOURCE POWER, WATTS:	200.0000	200.0000	200.0000	200.0000
SOURCE BEAM, DEG:	40.0000	40.0000	40.0000	40.0000
RECEIVER:	VIDICON	VIDICON	VIDICON	VIDICON
RECEIVER BEAM, DEG:	40.0000	40.0000	40.0000	40.0000
TAU:	.9000	.9000	.9000	.9000
F-NUMBER:	1.5000	1.5000	1.5000	1.5000
RANGE, FT:	24.4023	24.4023	29.7573	33.2921
RANGE, AL:	1.8742	1.8742	2.2855	2.5570
MINIMUM, FT:	2.5837	2.5837	3.0139	3.2687
MAXIMUM, FT:	.0000	.0000	.0000	.0000
BEST, FT:	24.4023	24.4023	29.7573	33.2921
AV LINES AT CTR:	393.9003	393.9003	322.7560	288.1938
AT EDGE:	393.9003	393.9003	322.7560	271.0501
CONTRAST AT CTR:	.1640	.1640	.1536	.1469
AT EDGE:	.0752	.0752	.0659	.0601
AV L.H.S. SW WIDTH, FT:	8.8196	8.8196	10.7119	11.9459
ACTUAL, FT:	8.2230	8.2230	10.1408	11.3969
- LIMITED BY:	R. ANG	R. ANG	R. ANG	R. ANG
AV R.H.S. SW WIDTH, FT:	8.9522	8.9522	10.9773	12.3444
ACTUAL, FT:	8.2230	8.2230	10.1408	10.2479
- LIMITED BY:	S. ANG	S. ANG	S. ANG	BKSKTR
AV SWATH WIDTH, FT:	17.7719	17.7719	21.6893	24.2904
ACTUAL, FT:	16.4460	16.4460	20.2816	21.6448
BEST, FT:	16,4460	16.4460	20.2816	21.6448

Figure 4. Final summary.

measure of the total horizontal distance over which satisfactory imagery is possible. **BEST, FT** is the maximum actual swath width. Zero will appear for this quantity unless the maximum swath width option has been selected.

3.0 BASIS OF SCOUTS CALCULATIONS

This section will outline the main features of the SCOUTS performance analysis and relate them to the corresponding steps in HUISID. This information is not strictly necessary for the general SCOUTS user, but it is useful to those who might desire to alter some of the program's internal parameters to suit their specific problems. To that end, the various water and hardware characteristics used by SCOUTS will be reviewed in Sections 3.1 and 3.2. Section 3.3 will provide an outline of the calculational flow keyed to the labeled line numbers of the detailed listing in the Appendix.

3.1 WATER CHARACTERISTICS

All the water characteristics used by SCOUTS are documented and referenced in HUISID and appear as stored matrices entered via DATA statements in the program MAIN. The following discussion of the program's features requires reference to the appendix, where the subroutines are arranged in alphabetical order. MAIN is the main program; the rest are subroutines. The function of each is explained in Section 4.

The array A(I, J) (line MAIN 21) stores the attenuation coefficient for coastal (I=1) and deep (I=2) water for forward (J=1) and backscattered (J=2) light. It was obtained from HUISID (Table 6.12). The array B(I,J,K) (MAIN 24) contains the ratio of the effective to actual attenuation coefficient indexed according to water type (I), forward or reverse propagation (J) and full beam angle (K); these data also come from Table 6.12 in HUISID. The interpolation array of corresponding beam angles is stored in matrix T (MAIN 49). Scattering coefficients, which were also obtained from Table 6.12, are stored in matrix S (MAIN 46). For both coastal and deep waters a rear hemisphere scattering (i.e., backscattering) percentage of 2 percent is used (line TVFU 33).

3.2 HARDWARE CHARACTERISTICS

The spectral characteristics of the source, receiver and water enter into the HUISID systems performance analysis through the constructed functions $G(\lambda)$ and $H(\lambda)$, which refer to forward and backscattered light, respectively. These functions are stored in the SCOUTS arrays G(I,J,K) and H(I,J,K) where I, J, K refer to water, receiver and source types, respectively. These entires were computed by subtracting entries in Table 6.8 of HUISID in accordance with the effective bandwidths listed in Table 6.5.

The imaging capability of the two types of cameras are stored in the array FM (I, J) (MAIN 30), where I refers to receiver type (i.e. I = 1 = vidicon and I = 2 = SIT) and J indexes two constants per receiver type. For a given camera tube current, i, the number of resolved TV lines, N, is obtained according to

$$N = \frac{\lg(i/FM(I, 1))}{FM(I, 2)} . \quad (3)$$

The values of $FM(I, J)$ were obtained by fitting Eq. (3) to available camera tube data. For the SIT the data were taken from HUISD, Figure 6.4(E), and the vidicon data from Reference 2. In each case the 50-percent contrast curve was used.

3.3 PERFORMANCE CALCULATIONS

The first calculation SCOUTS performs for each system's performance analysis is the evaluation of the minimum range. This occurs in lines HLIM 25-47. HLIMIT sets the minimum range equal to the greatest of (1) the source-receiver separation, (2) the minimum range which gives the required depth of field and (3) the minimum range which precludes the source and receiver looking at each other.

All the other calculations are done in the subroutine TVFUNC and the routines called by it. After some initialization, TVFUNC sets the range equal to that desired for single-range calculation (line TVFU 53) or to 110 percent of the minimum range in the maximizing modes (line TVFU 55). Next, a check is performed to make sure the display limit is satisfied (TVFU 59-64). If not, the remaining calculations are skipped and a flag is set so that a warning that the display limit has been exceeded is printed in the summary.

In lines TVFU 72-120, SCOUTS evaluates the number of lines at the center and the edge of the field of view. If less than the minimum (i.e. eight) number of lines is obtained at the center of the field of view, SCOUTS assigns minimum left- and right-power-limited swath widths (TVFU 101-105). If there is no power limit even at the edge of the field of view, the maximum swath widths are associated with the power limit (TVFU 117-119). In the case where the eight-line limit is encountered at some angle other than the maximum or minimum, this angle is found in an iterative search (TVFU 87-99) and the appropriate swath widths are computed (TVFU 109-114). In every case, the photocathode current is obtained in subroutine DFUNC, which implements Eqs. 6.17 and 6.34 of HUISD. EFUNC evaluates the number of resolved TV lines, using the matrix $FM(I, J)$.

The effects of backscatter are accounted for in lines TVFU 121-127. Using the subroutine XFUNC, which evaluates the photocathode current due to backscattered light (Eq. 6.48 of HUISD), the contrast is calculated at the center and at the right-hand edge of the field of view. Using the interpolation routine, GFUNC, an appropriate right-hand swath width due to backscatter, $W(7)$, is obtained. The geometric limitations on the swath width are next obtained ($W(2)$, $W(3)$, $W(5)$, $W(6)$) through the use of UFUNC and VFUNC. FNK finds the limiting right and left swath widths and descriptors. The limiting swath widths are summed to yield the total actual swath width.

Lines TVFU 146-184 determine whether the calculation is for a single-range or a maximizing calculation. If it is for a single-range calculation, TVFUNC returns control to HLIMIT. If not, TVFUNC determines whether the calculation is complete. If the last range increment is less than 5 percent of an attenuation length, control is returned to HLIMIT. If not, the range is incremented or decremented depending on the previous value of the swath width, and the performance is recalculated starting at line TVFU 40.

4.0 PROGRAM STRUCTURE

4.1 SUBROUTINE ORGANIZATION

SCOUTS consists of a main program and seventeen subroutines. The main program defines constants used in performing the calculations and acts as a program driver. Eleven of the seventeen subroutines perform the computations. The remaining six subroutines – SUBROUTINE DATAIN, SUBROUTINE CONVT, SUBROUTINE HLIMIT, SUBROUTINE UPDATE, SUBROUTINE DATAOT, and SUBROUTINE SUMMARY – were developed to provide flexibility of input and output. Figure 5 illustrates program hierarchy. A printout of each subroutine and of the main program is provided in the appendix.

SUBROUTINE DATAIN reads user-supplied input data and sets program control flags. If batch processing is desired, the user-supplied data are input via the card reader (unit number 5). Beginning in column one, the first card of this data deck must contain the five-character alphanumeric word BATCH. This card sets the flag for batch processing. Subsequent data cards are then read in an A6, 3X, 3F10.0 format. The alphanumeric information is a mnemonic instruction that directs the program to a specific section of code. The data values, where applicable, represent the initial, incremental, and final values of the data variables. Default values are set prior to reading any user-supplied data.

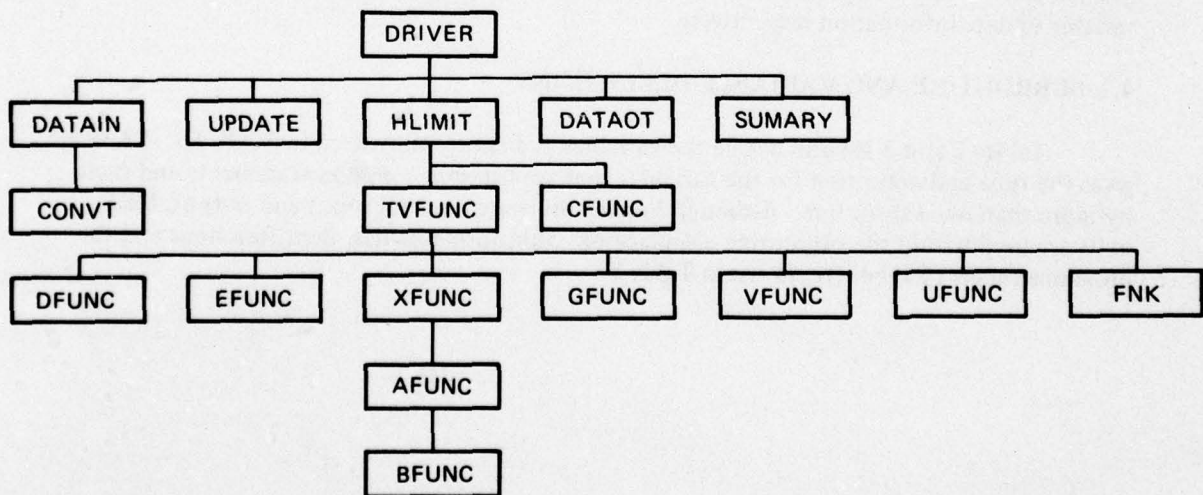


Figure 5. Program hierarchy.

If the program is being executed via a demand terminal, the terminal itself is the input device. In this case, interactive communication with the program is performed. The format of the input data, unless otherwise specified by the program, is G10.0. It should be noted that the program can be executed via a demand terminal without interactive communication by typing the word BATCH as the initial response. Data are input as though batch processing were being performed. (For more detailed information on input data, see the section under program operation.)

SUBROUTINE HLIMIT computes the minimum range and establishes the type of computations that will be made – single range, maximum range and/or maximum width.

Incrementing of the input data variables is performed by **SUBROUTINE UPDATE**. This subroutine is called only when parametric variables have been defined.

Two subroutines handle the printing of output data. Whenever execution is performed via a demand terminal, and no parametric variables have been defined, **SUBROUTINE DATAOT** prints the results after each evaluation is performed. This subroutine also writes a summary of the performance evaluation to the temporary disk file (unit number 9). At the conclusion of execution, **SUBROUTINE SUMMARY** reads the information stored on disk and prints tabular summaries of all performance evaluations computed. For batch processing, the printer (unit number 6) is the peripheral device utilized for output; for demand terminal processing, the terminal is the output device.

SUBROUTINE CONVT was developed to allow the demand terminal user to input integer values left-justified. If a two-digit integer value has been entered, the second digit is read as alphanumeric data and "converted" to the appropriate numeric value.

Data communication within the program is handled primarily through labeled common blocks. The four blocks established – **OPTS**, **TVCOM**, **IOLIST**, and **UPD** – store user-defined control options, program computing constants, input/output variables, and input variable update information respectively.

4.2 SUBROUTINE AND VARIABLE DEFINITIONS

Tables 2 and 3 list and define the variables and subroutines used in SCOUTS. Table 2 gives the type and definition for the variables that are listed in common statements and used by more than one subroutine. Although English units are used for input and output, MKS units are used within the subroutine calculations. Subroutine names, their functions and the programs called by them are shown in Table 3.

Table 2. Variable names and definitions.

Descriptive Variable	Type	Definition
NVIRON	integer	water type
WAVEHT	real	wave height
BTMRUF	real	bottom roughness
TARGLN	real	target length
TARGHT	real	target height
HEIGHT	real	height (or range)
SRSEP	real	source-receiver separation
SPOWR	real	source power
SDELF	real	full source angle
IRTP	integer	receiver type
RDELF	real	full receiver angle
OPTRAN	real	optics transmission
FNUM	real	f-number
ISTPE	integer	source type
RANGMN	real	minimum range
RANGMX	real	maximum range
RANGB	real	best range
SWWTHB	real	best swath width
HTUS	real	usual height
SWWTH	real	actual swath width
SWWTHL	real	actual L.H.S. swath width
SWWTHR	real	actual R.H.S. swath width
LIMFLG	integer	display limit flag
NDEXWL	integer	L.H.S. width index
NDEXWR	integer	R.H.S. width index
LINCEN	real	available lines at center
LINEDG	real	available lines at edge
CTRCEN	real	contrast at center
CTREDG	real	contrast at edge

Table 3. SCOUTS subroutines.

Subprogram	Function	Reference	Referenced by
MAIN	program driver; defines computing constants	DATAIN, HLIMIT, DATAOT, UPDATE, SUMARY	-
DATAIN	reads input data; sets control flags	CONVT	MAIN
HLIMIT	computes minimum range, defines type of computation	CFUNC, TVFUNC ATAN, TAN	MAIN
DATAOT	prints results after each evaluation; writes summary data to disk	-	MAIN
UPDATE	increments parametric input variables	-	MAIN
SUMARY	prints tabular summaries	-	MAIN
TVFUNC	main computational subroutines	DFUNC, EFUNC, VFUNC, XFUNC, GFUNC, UFUNC, FNK COS, ATAN, TAN	HLIMIT
CONVT	converts a single character to numeric data	-	DATAIN
XFUNC	calculates backscatter current	AFUNC	TVFUNC
AFUNC	Simpson's rule integration routine for backscatter integral	BFUNC, EXP, ALOG	XFUNC
BFUNC	integrand for backscatter integral	EXP	AFUNC
CFUNC	finds effective attenuation coefficient by interpolation	-	HLIMIT
DFUNC	computes signal current	TAN, ATAN, COS, EXP	TVFUNC
EFUNC	evaluates number of lines	ALOG	TVFUNC
FNK	interpolates for limiting width and description	-	TVFUNC
GFUNC	interpolates for backscatter width	TAN, ALOG	TVFUNC
UFUNC	calculates l.h.s. receiver and r.h.s. source widths	-	TVFUNC
VFUNC	calculates r.h.s. receiver and l.h.s. source widths	-	TVFUNC

5.0 PROGRAM OPERATION

This section is intended to provide the user with information that is required for successful execution of the visual search system program.

5.1 PRELIMINARY REQUIREMENTS

The visual search system program utilizes three peripheral devices for transmission of data: the card reader, the printer, and a disk. The card reader and the printer are assigned the standard FORTRAN unit numbers 5 and 6 respectively. Unit 9 is assigned to a disk file used temporarily to store unformatted data of summary output information. As required by the individual operating system involved, these unit numbers must be assigned to the appropriate peripheral device prior to compilation of the program.

After having assigned unit numbers to the appropriate peripheral device, the program is compiled and linked to create a set of absolute (or executable) binaries. Figure 6 illustrates the procedure used on the UNIVAC 1110 to create the absolute element for execution. In the example the absolutes are stored under the name VSEARCH of the file TVSENS. The program driver and each subprogram are stored under separate file element names.

```
@ FOR, N TVSENS. MAIN
@ FOR, N TVSENS. DATAIN
@ FOR, N TVSENS. CONVT
@ FOR, N TVSENS. HLIMIT
@ FOR, N TVSENS. TVFUNC
@ FOR, N TVSENS. UPDATE
@ FOR, N TVSENS. DATAOT
@ FOR, N TVSENS. SUMARY
@ FOR, N TVSENS. AFUNC
@ FOR, N TVSENS. BFUNC
@ FOR, N TVSENS. CFUNC
@ FOR, N TVSENS. DFUNC
@ FOR, N TVSENS. EFUNC
@ FOR, N TVSENS. FNK
@ FOR, N TVSENS. GFUNC
@ FOR, N TVSENS. UFUNC
@ FOR, N TVSENS. VFUNC
@ FOR, N TVSENS. XFUNC
@ PACK TVSENS.
@ PREP TVSENS.
@ MAP, N , TVSENS. VSEARCH
  IN TVSENS.
  END
```

Figure 6. Creation of absolute element.

5.2 BATCH PROCESSING

Batch processing of the absolute element VSEARCH is performed whenever the user does not wish interactive communication with the program. If batch processing is desired, the first card image of input data must be the Hollerith string BATCH. This five-character string informs the program that subsequent input will be in an A6, 3X, 3F10.0 format. The alphanumeric information is a mnemonic instruction that directs the program to a specific section of code. The data values, where applicable, represent the initial, the incremental, and the final values of the appropriate input variable. Table 4 lists the mnemonic instructions, the variable defined, and their default values. It should be noted that although the data assignment instructions may occur in any order within the data deck, the STOP instruction must follow any given set of data assignment instructions. The last card image of the data deck must be the mnemonic instruction FINISH. Figure 7 illustrates a sample deck setup for execution of VSEARCH through batch processing.

Table 4. Batch mnemonics.

Mnemonic	Defines	Default Value
NVIRON	=1., coastal; =2., deep	1.
ISTPE	=1., incandescent; =2., thallium; =3. mercury vapor	1.
IRTPE	=1., vidicon; =2., SIT	1.
WAVEHT	wave height	0.
BTMRUF	bottom roughness	0.
TARGLN	target length	0.
TARGHT	target height	0.
SRSEP	source-receiver separation	0.
SPOWR	source power	0.
SDELF	full source angle	0.
RDELF	full receiver angle	0.
OPTRAN	optic transmission	0.
FNUM	f-number	.5
HEIGHT	range (or height)	0.
MAXSW	sets flag to compute maximum width	—
MAXRAN	sets flag to compute maximum range	—
SINGRN	sets flag to compute single range	—
STOP	stop reading input; end-of-record indicator for input	—
FINISH	stop execution; end-of-file indicator for input	—

BATCH			
NVIRON	1.00000	.00000	.00000
WAVEHT	2.00000	.00000	.00000
BTMRUF	.50000	.00000	.00000
TARGLN	10.00000	.00000	.00000
TARGHT	.50000	.00000	.00000
SRSEP	1.00000	.00000	.00000
ISTPE	2.00000	.00000	.00000
SPOWR	200.00000	.00000	.00000
SDELF	40.00000	.00000	.00000
IRTPE	1.00000	.00000	.00000
RDELF	40.00000	.00000	.00000
OPTRAN	.90000	.00000	.00000
FNUM	1.50000	.00000	.00000
MAXSW	.00000	.00000	.00000
STOP	.00000	.00000	.00000
NVIRON	2.00000	.00000	.00000
RANGE	65.00000	10.00000	105.00000
MAXSW	.00000	.00000	.00000
STOP	.00000	.00000	.00000
SRSEP	1.00000	1.00000	4.00000
RANGE	.00000	.00000	.00000
MAXSW	.00000	.00000	.00000
STOP	.00000	.00000	.00000
SRSEP	5.00000	.00000	.00000
SPOWR	50.00000	50.00000	105.00000
MAXSW	.00000	.00000	.00000
STOP	.00000	.00000	.00000
SDELF	20.00000	.00000	.00000
RDELF	20.00000	.00000	.00000
MAXSW	.00000	.00000	.00000
STOP	.00000	.00000	.00000
FINISH	.00000	.00000	.00000

Figure 7. Sample deck setup for batch execution of VSEARCH.

5.3 DEMAND TERMINAL USE

SCOUTS should be executed via a demand terminal whenever the user wishes interactive communication. A demand run also provides intermediate information that can be useful in determining which input variables should be changed.

Execution of the absolute element is performed by responding to the program's questions. Questions concerning environment, source type and receiver type are answered in an I1 format. Other initial input is in a G10.0 format. After the initial input, SCOUTS asks for direction as to parametric or nonparametric, the calculational mode, whether another run is desired, etc. These questions are also answered in an I1 format.

It should be noted that batch processing of SCOUTS can be performed from a demand terminal by responding 'BATCH' to the first question asked. In this case the succeeding input will have to follow exactly the format of Figure 7.

REFERENCES

1. Naval Undersea Center. NUC TP 303, Handbook of Underwater Imaging System Design, by C. J. Funk, S. B. Bryant and P. J. Heckman, Jr. July 1972.
2. Biberman, L. M. and S. Nudelman. Photoelectronic Imaging Devices, Vol. 2, p. 539. Plenum Press, New York, N. Y., 1971.
3. RCA Commercial Engineering. RCA Electro-Optics Handbook, p. 121. Harrison, N. J., 1974.

APPENDIX

This appendix contains a complete FORTRAN IV listing of SCOUTS, arranged in alphabetical order by subroutine name. MAIN is the main program; all the rest are FORTRAN subroutines. All programs have compiled successfully on the NOSC UNIVAC 1110.

000001	001	FUNCTION AFUNC(A)	AFUN 1
000002	001		AFUN 2
000003	001	C	AFUN 3
000004	001	U3=-1.*ALOG(.00125*(EXP(-A)-EXP(-2.*A))/A)/A	AFUN 4
000005	001	U1=n3	AFUN 5
000006	001	10 U2=n3-ALOG(n1)/A	AFUN 6
000007	001	IF (ABS((D2-U1)/U2).LT..01) GO TO 20	AFUN 7
000008	001	U1=n2	AFUN 8
000009	001	GO TO 10	AFUN 9
000010	001	20 U0=(U2-1.)/2.	AFUN 10
000011	001	U1=n0/3.	AFUN 11
000012	001	U4=nFUNC(A+1.)*RFUNC(A+02)	AFUN 12
000013	001	U7=nFUNC(A+1.*U0)	AFUN 13
000014	001	U2=n1*(n4+4.*D7)	AFUN 14
000015	001	U3=1.	AFUN 15
000016	001	30 U3=p.*U3	AFUN 16
000017	001	U1=n1/2.	AFUN 17
000018	001	U0=n0/2.	AFUN 18
000019	001	D9=nFUNC(A+1.*U0)	AFUN 19
000020	001	U8=n9	AFUN 20
000021	001	E1=EXP(-1.*A*2.*U0)	AFUN 21
000022	001	J1=n3-1.	AFUN 22
000023	001	U0 40 J=1,J1	AFUN 23
000024	001	E2=(1.+(2.*FLOAT(J)-1.)*D0)/(1.+(2.*FLOAT(J)+1.)*D0)	AFUN 24
000025	001	U8=n8+E1+E2	AFUN 25
000026	001	U9=n9+U8	AFUN 26
000027	001	40 CONTINUE	AFUN 27
000028	001	U5=n1*(n4+2.*D7+4.*n9)	AFUN 28
000029	001	IF (ABS((D5-U2)/U5).LT..01) GO TO 50	AFUN 29
000030	001	U2=n5	AFUN 30
000031	001	U7=n9+U7	AFUN 31
000032	001	GO TO 30	AFUN 32
000033	001	50 AFUNC=U5	AFUN 33
000034	001	RETURN	AFUN 34
000035	001	END	AFUN 35-

000001	001	FUNCTION RFUNC(A,R)	RFUN 1
000002	001	RFUNC=(EXP(-A*R))/(R*R)	RFUN 2
000003	001	RETURN	RFUN 3
000004	001	END	RFUN 4-

000001	001	FUNCTION CFUNC(TUIR,THEY,TWAT)	CFUN 1
000002	001	C	CFUN 2
000003	001	C	CFUN 3
000004	001	COMMON BLOCK OF COMPUTING CONSTANTS	CFUN 4
000005	001	COMMON /TVCOM/ A(2,2), R(2,11,2), C(20),U(2,2), E(10),	CFUN 5
000006	001	1 F(12,2), G(2,2,3), H(2,2,3), N(10), O(15), P(2,2),	CFUN 6
000007	001	2 Q(15), S(2), T(11), W(10), Z(2), PII	CFUN 7
000008	001	C	CFUN 8
000009	001	DO 10 IT=1,11	CFUN 9
000010	001	IF (THEY.LT.T(IT)) GO TO 20	CFUN 10
000011	001	10 CONTINUE	CFUN 11
000012	001	20 ID2=MIN0(IT,11)	CFUN 12
000013	001	IT=ID2-1	CFUN 13
000014	001	U5=R(TWAT,IT,TUIR)	CFUN 14
000015	001	CFUNC=U5*(THEY-T(IT))/(T(ID2)-T(IT))*(B(TWAT,ID2,TUIR)-R(TWAT,IT	CFUN 15
000016	001	1,TUIR))	CFUN 16
000017	001	RETURN	CFUN 17-
	001	END	

000001	001	SUBROUTINE CONV (IVAL)	CONV 1
000002	001	C	CONV 2
000003	001	C	CONV 3
000004	001	C	CONV 4
000005	001	C	CONV 5
000006	001	C	CONV 6
000007	001	C	CONV 7
000008	001	C	CONV 8
000009	001	C	CONV 9
000010	001	C	CONV 10
000011	001	C	CONV 11
000012	001	C	CONV 12
000013	001	C	CONV 13
000014	001	C	CONV 14
000015	001	C	CONV 15
000016	001	C	CONV 16
000017	001	C	CONV 17
000018	001	C	CONV 18
000019	001	C	CONV 19
000020	001	C	CONV 20
000021	001	C	CONV 21-

000001	001	SUBROUTINE DATAIN (ISTART,ISUM)	DAIN 1
000002	001	C	DAIN 2
000003	001	C	DAIN 3
000004	001	C	DAIN 4
000005	001	C	DAIN 5
000006	001	C	DAIN 6
000007	001	C	DAIN 7
000008	001	C	DAIN 8
000009	001	C	DAIN 9
000010	001	C	DAIN 10
000011	001	C	DAIN 11
000012	001	C	DAIN 12
000013	001	C	DAIN 13
000014	001	C	DAIN 14
000015	001	C	DAIN 15
000016	001	C	DAIN 16
000017	001	C	DAIN 17
000018	001	C	DAIN 18
000019	001	C	DAIN 19
000020	001	C	DAIN 20
000021	001	C	DAIN 21
000022	001	C	DAIN 22
000023	001	C	DAIN 23
000024	001	C	DAIN 24
000025	001	C	DAIN 25
000026	001	C	DAIN 26
000027	001	C	DAIN 27
000028	001	C	DAIN 28
000029	001	C	DAIN 29
000030	001	C	DAIN 30
000031	001	C	DAIN 31
000032	001	C	DAIN 32
000033	001	C	DAIN 33
000034	001	C	DAIN 34
000035	001	C	DAIN 35
000036	001	C	DAIN 36
000037	001	C	DAIN 37
000038	001	C	DAIN 38
000039	001	C	DAIN 39
000040	001	C	DAIN 40
000041	001	C	DAIN 41
000042	001	C	DAIN 42
000043	001	C	DAIN 43
000044	001	C	DAIN 44
000045	001	C	DAIN 45
000046	001	C	DAIN 46
000047	001	C	DAIN 47
000048	001	C	DAIN 48
000049	001	C	DAIN 49
000050	001	C	DAIN 50
000051	001	C	DAIN 51
000052	001	C	DAIN 52
000053	001	C	DAIN 53
000054	001	C	DAIN 54
000055	001	C	DAIN 55
000056	001	C	DAIN 56
000057	001	C	DAIN 57
000058	001	C	DAIN 58
000059	001	C	DAIN 59
000060	001	C	DAIN 60
000061	001	C	DAIN 61
000062	001	C	DAIN 62
000063	001	C	DAIN 63
000064	001	C	DAIN 64
000065	001	C	DAIN 65
000066	001	C	DAIN 66
000067	001	C	DAIN 67
000068	001	C	DAIN 68
000069	001	C	DAIN 69
000070	001	C	DAIN 70
000071	001	C	DAIN 71
000072	001	C	DAIN 72
000073	001	C	DAIN 73
000074	001	C	DAIN 74
000075	001	C	DAIN 75
000076	001	C	DAIN 76
000077	001	C	DAIN 77
000078	001	C	DAIN 78
000079	001	C	DAIN 79
000080	001	C	DAIN 80
000081	001	C	DAIN 81
000082	001	C	DAIN 82
000083	001	C	DAIN 83
000084	001	C	DAIN 84
000085	001	C	DAIN 85

000086	001	PRINT 650	DAIN 86
000087	001	80 PRINT 660	DAIN 87
000088	001	MEAN (5:540) SNEP	DAIN 88
000089	001	IF (SPSEF.LT.0.) GO TO 80	DAIN 89
000090	001	IF (ISTANT.F0-1) GO TO 240	DAIN 90
000091	001	PRINT 650	DAIN 91
000092	001	90 PRINT 660	DAIN 92
000093	001	MEAN (5:530) ISTPE	DAIN 93
000094	001	IF (ISTPE.LT.1.00.(ISTPE.GT.3) GO TO 90	DAIN 94
000095	001	IF (ISTANT.F0-1) GO TO 240	DAIN 95
000096	001	100 PRINT 670	DAIN 96
000097	001	MEAN (5:540) SPOWH	DAIN 97
000098	001	IF (SPOWH.LT.0.) GO TO 100	DAIN 98
000099	001	IF (ISTANT.F0-1) GO TO 240	DAIN 99
000100	001	110 PRINT 680	DAIN100
000101	001	MEAN (5:540) SUELF	DAIN101
000102	001	IF (SUELF.LT.0..00.SUELF.GT.100.) GO TO 110	DAIN102
000103	001	IF (ISTANT.F0-1) GO TO 240	DAIN103
000104	001	PRINT 690	DAIN104
000105	001	120 PRINT 700	DAIN105
000106	001	MEAN (5:530) INTPE	DAIN106
000107	001	IF (INTPE.NE.1.AND.INTPE.NE.2) GO TO 120	DAIN107
000108	001	IF (ISTANT.F0-1) GO TO 240	DAIN108
000109	001	130 PRINT 710	DAIN109
000110	001	MEAN (5:540) RUELF	DAIN110
000111	001	IF (RUELF.LT.0..00.RUELF.GT.100.) GO TO 130	DAIN111
000112	001	IF (ISTANT.F0-1) GO TO 240	DAIN112
000113	001	140 PRINT 720	DAIN113
000114	001	MEAN (5:540) OPTMAN	DAIN114
000115	001	IF (OPTMAN.LT.0..00.OPTMAN.GT.1.) GO TO 140	DAIN115
000116	001	IF (ISTANT.F0-1) GO TO 240	DAIN116
000117	001	150 PRINT 730	DAIN117
000118	001	MEAN (5:540) FNMH	DAIN118
000119	001	IF (FNMH.LT..5) GO TO 150	DAIN119
000120	001	IF (ISTANT.F0-1) GO TO 240	DAIN120
000121	001	C	DAIN121
000122	001	C	DAIN122
000123	001	C	DAIN123
000124	001	PARAMETRIC OR NON-PARAMETRIC MODE? IF NON-PARAMETRIC	DAIN124
000125	001	MODE (IPARAM.NF.1): VARIABLE INPUT COMPLETE.	DAIN125
000126	001	PRINT 740	DAIN126
000127	001	MEAN (5:530) IPANAM	DAIN127
000128	001	IF (IPARAM.NE.1) GO TO 200	DAIN128
000129	001	C	DAIN129
000130	001	C	DAIN130
000131	001	C	DAIN131
000132	001	GIVE INDEX OF SPECIFIC PARAMETRIC VARIABLE. THEN READ	DAIN132
000133	001	ITS INCREMENTAL AND ENDING VALUES.	DAIN133
000134	001	PRINT 750	DAIN134
000135	001	160 MEAN (5:530) IN*IN2	DAIN135
000136	001	IF (IN.F0-0) GO TO 180	DAIN136
000137	001	IF (IN2.EQ.10) GO TO 170	DAIN137
000138	001	CALC CONV (IN2)	DAIN138
000139	001	IN2=10*IN*IN2	DAIN139
000140	001	170 PRINT 760	DAIN140
000141	001	MEAN (5:540) STEP(IN)	DAIN141
000142	001	PRINT 770	DAIN142
000143	001	MEAN (5:540) EVAL(IN)	DAIN143
000144	001	180 IF (ISTANT.F0-1) GO TO 260	DAIN144
000145	001	C	DAIN145
000146	001	C	DAIN146
000147	001	C	DAIN147
000148	001	IF ADDITIONAL PARAMETRIC VARIABLES ARE DESIRED, CHANGE	DAIN148
000149	001	INDEX.	DAIN149
000150	001	PRINT 780	DAIN150
000151	001	MEAN (5:530) IN2	DAIN151
000152	001	IF (IN2.NE.1) GO TO 190	DAIN152
000153	001	PRINT 850	DAIN153
000154	001	MEAN (5:530) ICAL	DAIN154
000155	001	IF (ICAL.GT.4) ICAL=1	DAIN155
000156	001	IF (ISTANT.F0-1) GO TO 490	DAIN156
000157	001	C	DAIN157
000158	001	C	DAIN158
000159	001	C	DAIN159
000160	001	VARIABLE INPUT COMPLETE.	DAIN160
000161	001	200 PRINT 790	DAIN161
000162	001	ISTANT=1	DAIN162
000163	001	GO TO 490	DAIN163
000164	001	C	DAIN164
000165	001	C	DAIN165
000166	001	C	DAIN166
000167	001	IF PREVIOUS CALCULATIONS HAVE BEEN PERFORMED.	DAIN167
000168	001	C	DAIN168
000169	001	C	DAIN169
	001	IF ANOTHER RUN IS DESIRED, MAKE NECESSARY CHANGES IF ANY	DAIN169
	001	210 PRINT 790	
	001	MEAN (5:530) IN2	

000170	001	IF (INZ.NF.1) GO TO 490	DAIN170
000171	001	IF (IPARAM.FG.1) CALL SIMMAY (IRUN,ISUM)	DAIN171
000172	001	ISTART=1	DAIN172
000173	001	PRINT /40	DAIN173
000174	001	READ (5,530) IPARAM	DAIN174
000175	001	PRINT #00	DAIN175
000176	001	READ (5,530) IN2	DAIN176
000177	001	IF (INZ.NF.1) GO TO 490	DAIN177
000178	001	C	DAIN178
000179	001	C	DAIN179
000180	001	C	DAIN180
000181	001	PRINT /50	DAIN181
000182	001	220 READ (5,530) IN=IN2	DAIN182
000183	001	IF (INZ.EQ.1H) GO TO 230	DAIN183
000184	001	CALL COMVT (IN2)	DAIN184
000185	001	IN=10*IN*IN2	DAIN185
000186	001	230 GO TO (30,40,120,40,50,60,70,80,100,110,130,140,150,240), IN	DAIN186
000187	001	C	DAIN187
000188	001	C	DAIN188
000189	001	C	DAIN189
000190	001	240 IF (IPARAM.NE.1) GO TO 260	DAIN190
000191	001	PRINT #20	DAIN191
000192	001	READ (5,530) IN2	DAIN192
000193	001	IF (INZ.NF.1) GO TO 250	DAIN193
000194	001	GO TO 170	DAIN194
000195	001	250 STEP(IN)=0.	DAIN195
000196	001	C	DAIN196
000197	001	C	DAIN197
000198	001	260 PRINT #10	DAIN198
000199	001	READ (5,530) IN2	DAIN199
000200	001	IF (INZ.NF.1) GO TO 190	DAIN200
000201	001	PRINT #30	DAIN201
000202	001	GO TO 220	DAIN202
000203	001	C	DAIN203
000204	001	C	DAIN204
000205	001	C	DAIN205
000206	001	C	DAIN206
000207	001	C	DAIN207
000208	001	C	DAIN208
000209	001	270 READ (5,530) IVAR,21,22,23	DAIN209
000210	001	WRITE (6,550) IVAR,21,22,23	DAIN210
000211	001	C	DAIN211
000212	001	IF (21.LT.0.) 21=0.	DAIN212
000213	001	IF (ARS(2).EQ.0.) 23=21	DAIN213
000214	001	IF (ARS(2).NE.0.) IPARAM=1	DAIN214
000215	001	C	DAIN215
000216	001	C	DAIN216
000217	001	C	DAIN217
000218	001	UN 280 I=1,19	DAIN218
000219	001	IP=1	DAIN219
000220	001	IF (IVAR.FG.INST(1)) GO TO 290	DAIN220
000221	001	280 CONTINUE	DAIN221
000222	001	GO TO 270	DAIN222
000223	001	C	DAIN223
000224	001	290 GO TO (300,360,390,310,320,330,340,350,370,380,400,410,420,470,4300	DAIN224
000225	001	1,440,460,480,490), IP	DAIN225
000226	001	C	DAIN226
000227	001	C	DAIN227
000228	001	300 NVIRON=1	DAIN228
000229	001	IF (21.FG.2.) NVIRON=2	DAIN229
000230	001	STEP(1)=22	DAIN230
000231	001	ENVAL(1)=23	DAIN231
000232	001	GO TO 270	DAIN232
000233	001	C	DAIN233
000234	001	C	DAIN234
000235	001	310 WAVEHT=71	DAIN235
000236	001	STEP(4)=22	DAIN236
000237	001	ENVAL(4)=23	DAIN237
000238	001	GO TO 270	DAIN238
000239	001	C	DAIN239
000240	001	C	DAIN240
000241	001	320 HTMOUF=71	DAIN241
000242	001	STEP(5)=22	DAIN242
000243	001	ENVAL(5)=23	DAIN243
000244	001	GO TO 270	DAIN244
000245	001	C	DAIN245
000246	001	C	DAIN246
000247	001	330 TARGLN=71	DAIN247
000248	001	STEP(6)=22	DAIN248
000249	001	ENVAL(6)=23	DAIN249
000250	001	GO TO 270	DAIN250
000251	001	C	DAIN251
000252	001	C	DAIN252
000253	001	340 TARGHT=71	DAIN253
000254	001	STEP(7)=22	DAIN254

000255	001	EDVAL(7)=Z3	DAIN255
000256	001	GO TO 270	DAIN256
000257	001	C	DAIN257
000258	001	C SOURCE RECEIVER SEPARATION	DAIN258
000259	001	350 SRSEP=Z1	DAIN259
000260	001	STEP(1)=Z2	DAIN260
000261	001	EDVAL(8)=Z3	DAIN261
000262	001	GO TO 270	DAIN262
000263	001	C	DAIN263
000264	001	C SOURCE TYPE	DAIN264
000265	001	360 ISIPE=1	DAIN265
000266	001	IF (Z1.FU.2.) ISIPE=2	DAIN266
000267	001	IF (Z1.FU.3.) ISIPE=3	DAIN267
000268	001	STEP(2)=Z2	DAIN268
000269	001	EDVAL(12)=Z3	DAIN269
000270	001	GO TO 270	DAIN270
000271	001	C	DAIN271
000272	001	C SOURCE POWER	DAIN272
000273	001	370 SPOWER=Z1	DAIN273
000274	001	STEP(9)=Z2	DAIN274
000275	001	EDVAL(14)=Z3	DAIN275
000276	001	GO TO 270	DAIN276
000277	001	C	DAIN277
000278	001	C FULL SOURCE ANGLE	DAIN278
000279	001	380 SDELF=Z1	DAIN279
000280	001	STEP(10)=Z2	DAIN280
000281	001	EDVAL(10)=Z3	DAIN281
000282	001	IF (SDELF.GT.180.) SDELF=SDELF-180.	DAIN282
000283	001	IF (ARS(Z2).EQ.0.) EDVAL(10)=SDELF	DAIN283
000284	001	GO TO 270	DAIN284
000285	001	C	DAIN285
000286	001	C RECEIVER TYPE	DAIN286
000287	001	390 IRIPE=1	DAIN287
000288	001	IF (Z1.FU.2.) IRIPE=2	DAIN288
000289	001	STEP(3)=Z2	DAIN289
000290	001	EDVAL(13)=Z3	DAIN290
000291	001	GO TO 270	DAIN291
000292	001	C	DAIN292
000293	001	C FULL RECEIVER ANGLE	DAIN293
000294	001	400 RDELF=Z1	DAIN294
000295	001	STEP(11)=Z2	DAIN295
000296	001	EDVAL(11)=Z3	DAIN296
000297	001	IF (RDELF.GT.180.) RDELF=RDELF-180.	DAIN297
000298	001	IF (ARS(Z2).EQ.0.) EDVAL(11)=RDELF	DAIN298
000299	001	GO TO 270	DAIN299
000300	001	C	DAIN300
000301	001	C TRANSMISSION OF OPTICS	DAIN301
000302	001	410 OPTRAN=Z1	DAIN302
000303	001	STEP(12)=Z2	DAIN303
000304	001	EDVAL(12)=Z3	DAIN304
000305	001	IF (OPTRAN.GT.1.) OPTRAN=0.	DAIN305
000306	001	IF (ARS(Z2).EQ.0.) EDVAL(12)=OPTRAN	DAIN306
000307	001	GO TO 270	DAIN307
000308	001	C	DAIN308
000309	001	C F=NUMHFN	DAIN309
000310	001	420 FNUM=Z1	DAIN310
000311	001	STEP(13)=Z2	DAIN311
000312	001	EDVAL(13)=Z3	DAIN312
000313	001	IF (FNUM.LT..5) FNUM=.5	DAIN313
000314	001	IF (ARS(Z2).EQ.0.) EDVAL(13)=FNUM	DAIN314
000315	001	GO TO 270	DAIN315
000316	001	C	DAIN316
000317	001	C SET FLAG TO CALCULATE MAXIMUM WIDTH	DAIN317
000318	001	430 IF (ICAL.FU.2) GO TO 450	DAIN318
000319	001	ICAL=1	DAIN319
000320	001	GO TO 270	DAIN320
000321	001	C	DAIN321
000322	001	C SET FLAG TO CALCULATE MAXIMUM RANGE	DAIN322
000323	001	440 IF (ICAL.EQ.3) GO TO 450	DAIN323
000324	001	ICAL=2	DAIN324
000325	001	GO TO 270	DAIN325
000326	001	C	DAIN326
000327	001	C SET FLAG TO CALCULATE BOTH MAXIMUM WIDTH AND	DAIN327
000328	001	MAXIMUM RANGE	DAIN328
000329	001	450 ICAL=4	DAIN329
000330	001	GO TO 270	DAIN330
000331	001	C	DAIN331
000332	001	C SET FLAG TO CALCULATE SINGLE RANGE ONLY	DAIN332
000333	001	460 ICAL=1	DAIN333
000334	001	GO TO 270	DAIN334
000335	001	C	DAIN335
000336	001	C RANGE	DAIN336
000337	001	470 HEIGHT=Z1	DAIN337
000338	001	STEP(14)=Z2	DAIN338
000339	001	EDVAL(14)=Z3	DAIN339
000340	001	GO TO 270	DAIN340


```

000341      001      C      SET FLAG TO PERFORM CALCULATIONS; ON      DAIN341
000342      001      C      RF-WRITE IS IN EFFECT - READ NEW SET OF CHANGES      DAIN342
000343      001      C      480 ISTART=1      DAIN343
000344      001      C      490 CONTINUE      DAIN344
000345      001      C      RETURN      DAIN345
000346      001      C      DAIN346
000347      001      C      DAIN347
000348      001      C      DAIN348
000349      001      C      DAIN349
000350      001      C      DAIN350
000351      001      500 FORMAT (46H IF HATCH RUN; ENTER - HATCH; OTHERWISE RETURN)      DAIN351
000352      001      510 FORMAT (A6.3X.3F10.0)      DAIN352
000353      001      520 FORMAT (1X.A6.4H RUN)      DAIN353
000354      001      530 FORMAT (11.A1)      DAIN354
000355      001      540 FORMAT (610.0)      DAIN355
000356      001      550 FORMAT (1X.A6.3X.3F15.5)      DAIN356
000357      001      560 FORMAT (27X.22HCONVENTIONAL TV SYSTEM/3Y.11HENVIRONMENT)      DAIN357
000358      001      570 FORMAT (26H COASTAL OR DEEP. 1 OR 2:?)      DAIN358
000359      001      580 FORMAT (32H WAVE HEIGHT-PEAK TO TROUGH;FT:?)      DAIN359
000360      001      590 FORMAT (37H BOTTOM ROUGHNESS-PEAK TO TROUGH;FT:?)      DAIN360
000361      001      600 FORMAT (3X.17HTARGET DIMENSIONS)      DAIN361
000362      001      610 FORMAT (22H LENGTH OF TARGET;FT:?)      DAIN362
000363      001      620 FORMAT (22H HEIGHT OF TARGET;FT:?)      DAIN363
000364      001      630 FORMAT (3X.24HPLATFORM CHARACTERISTICS)      DAIN364
000365      001      640 FORMAT (32H SOURCE RECEIVER SEPARATION;FT:?)      DAIN365
000366      001      650 FORMAT (3X.22HSOURCE CHARACTERISTICS)      DAIN366
000367      001      660 FORMAT (61H INCANDESCENT; THALLIUM IODIDE; OR MERCURY VAPOR; 1, 2      DAIN367
000368      001      1OR 1:?)      DAIN368
000369      001      670 FORMAT (21H SOURCE POWER;WATTS:?)      DAIN369
000370      001      680 FORMAT (24H FULL SOURCE ANGLE;DEG:?)      DAIN370
000371      001      690 FORMAT (3X.24HRECEIVER CHARACTERISTICS)      DAIN371
000372      001      700 FORMAT (25H VIDEOCON OR SIT. 1 OR 2:?)      DAIN372
000373      001      710 FORMAT (26H FULL RECEIVER ANGLE;DEG:?)      DAIN373
000374      001      720 FORMAT (25H TRANSMISSION OF OPTICS:?)      DAIN374
000375      001      730 FORMAT (11H F-NUMBER:?)      DAIN375
000376      001      740 FORMAT (44H PARAMETRIC OR NON-PARAMETRIC MODE. 1 OR 2:?)      DAIN376
000377      001      750 FORMAT (46H WATER TYPE(1);SOURCE TYPE(2);RECEIVER TYPE(3);52H WAVE      DAIN377
000378      001      1 HEIGHT(4);BOTTOM ROUGHNESS(5);TARGET LENGTH(6);51H TARGET HEIGHT(      DAIN378
000379      001      2);4-R SEPARATION(8);SOURCE POWER(9);36H SOURCE ANGLE(10);RECEIVER      DAIN379
000380      001      3 ANGLE(11);47H OPTICS TRANSMISSION(12);F-NUMBER(13);RANGE(14);6H      DAIN380
000381      001      4 ANGLE:?)      DAIN381
000382      001      760 FORMAT (40H INCREMENTAL VALUE (IN FLOATING POINT):?)      DAIN382
000383      001      770 FORMAT (35H ENDING VALUE (IN FLOATING POINT):?)      DAIN383
000384      001      780 FORMAT (25H PARAMETER INPUT COMPLETE?)      DAIN384
000385      001      790 FORMAT (45H DO YOU WANT ANOTHER RUN; YES(1) OR NO(2):?)      DAIN385
000386      001      800 FORMAT (39H CHANGE PARAMETER(S); YES(1) OR NO(2):?)      DAIN386
000387      001      810 FORMAT (32H MORE CHANGES; YES(1) OR NO(2):?)      DAIN387
000388      001      820 FORMAT (44H PARAMETRIC OR NON-PARAMETRIC VARIABLE. 1 OR 2:?)      DAIN388
000389      001      830 FORMAT (14H CHANGE INDEX)      DAIN389
000390      001      840 FORMAT (45H MORE PARAMETRIC VARIABLES; YES(1) OR NO(2):?)      DAIN390
000391      001      850 FORMAT (54H MAX. RANGE(2); MAX. WIDTH(3); BOTH(4) OR NEITHER(5):?)      DAIN391
000392      001      END      DAIN392

```

```

000001      001      SUBROUTINE DATA01 (ISUM)      DAOT 1
000002      001      C      DAOT 2
000003      001      C      DAOT 3
000004      001      C      DAOT 4
000005      001      C      DAOT 5
000006      001      C      DAOT 6
000007      001      C      DAOT 7
000008      001      C      DAOT 8
000009      001      C      DAOT 9
000010      001      C      DAOT 10
000011      001      C      DAOT 11
000012      001      C      DAOT 12
000013      001      C      DAOT 13
000014      001      C      DAOT 14
000015      001      C      DAOT 15
000016      001      C      DAOT 16
000017      001      C      DAOT 17
000018      001      C      DAOT 18
000019      001      C      DAOT 19
000020      001      C      DAOT 20
000021      001      C      DAOT 21
000022      001      C      DAOT 22
000023      001      C      DAOT 23
000024      001      C      DAOT 24
000025      001      C      DAOT 25
000026      001      C      DAOT 26
000027      001      C      DAOT 27
000028      001      C      DAOT 28
000029      001      C      DAOT 29
000030      001      C      DAOT 30

```

000031	001	NAME1=0HLOASTA	DAOT 31
000032	001	NAME2=0HIL	DAOT 32
000033	001	IF (NVIPUN.EQ.1) GO TO 20	DAOT 33
000034	001	NAME1=0HUFFP	DAOT 34
000035	001	NAME2=0H	DAOT 35
000036	001	20 NAME3=0HINLAND	DAOT 36
000037	001	NAME4=0H.	DAOT 37
000038	001	IF (ISIPLE=2) 50,30,40	DAOT 38
000039	001	30 NAME5=0HITHAL.	DAOT 39
000040	001	NAME4=0H100.	DAOT 40
000041	001	GO TO 50	DAOT 41
000042	001	40 NAME5=0HMEHC.	DAOT 42
000043	001	NAME4=0HVA.	DAOT 43
000044	001	50 NAME5=0HVIDICO	DAOT 44
000045	001	NAME6=0HNN	DAOT 45
000046	001	IF (IRIPLE.F0.1) GO TO 60	DAOT 46
000047	001	NAME5=0H5IT	DAOT 47
000048	001	NAME6=0H	DAOT 48
000049	001	C	DAOT 49
000050	001	60 PRINT 40, NAME1,NAME2,NAME3,NAME4	DAOT 50
000051	001	PRINT 100, WAVEHT,SPOWH	DAOT 51
000052	001	PRINT 110, RTMHUF,SNELF	DAOT 52
000053	001	PRINT 120, TARGLN,NAME5,NAME6	DAOT 53
000054	001	PRINT 130, TARGHT,NDLEF	DAOT 54
000055	001	PRINT 140, OPTHAN	DAOT 55
000056	001	PRINT 150, SNSEF,FNUM	DAOT 56
000057	001	PRINT 160, RANGE,HANGAL	DAOT 57
000058	001	PRINT 170, LINCEN,LINFUG	DAOT 58
000059	001	PRINT 180, CTRECN,CTHEDG	DAOT 59
000060	001	PRINT 190, AVSWL,SWWTHL	DAOT 60
000061	001	PRINT 200, TA(NDXWL)	DAOT 61
000062	001	PRINT 210, AVSWR,SWWTHR	DAOT 62
000063	001	PRINT 220, TA(NDXWR)	DAOT 63
000064	001	PRINT 230, AVSW,SWWTH,SWWTHB	DAOT 64
000065	001	C	DAOT 65
000066	001	70 IF (ISUM.F0.1) READIN 4	DAOT 66
000067	001	WRITE (9) NVIPUN,AVEHT,HTMRUF	DAOT 67
000068	001	WRITE (9) TARGLN,TARGHT	DAOT 68
000069	001	WRITE (9) SNSEF	DAOT 69
000070	001	WRITE (9) ISIPLE,SPOWH,SNELF	DAOT 70
000071	001	WRITE (9) IRIPLE,NDLEF,OPTHAN,FNUM	DAOT 71
000072	001	WRITE (9) RANGE,HANGAL,RANGMN,HANGMX,RANGEB	DAOT 72
000073	001	WRITE (9) LINCEN,LINCENB,CTHCEN,CTRENG	DAOT 73
000074	001	WRITE (9) AVSWL,SWWTHL,TA(NDXWL)	DAOT 74
000075	001	WRITE (9) AVSWR,SWWTHR,TA(NDXWR)	DAOT 75
000076	001	WRITE (9) AVSW,SWWTH,SWWTHB	DAOT 76
000077	001	C	DAOT 77
000078	001	C	DAOT 78
000079	001	DO 40 I=1,16	DAOT 79
000080	001	ZFN(I)=0.	DAOT 80
000081	001	80 CONTINUE	DAOT 81
000082	001	NDXWL=0	DAOT 82
000083	001	NDXWR=0	DAOT 83
000084	001	LINFLE=0	DAOT 84
000085	001	C	DAOT 85
000086	001	RETRN	DAOT 86
000087	001	C	DAOT 87
000088	001	C	DAOT 88
000089	001	C	DAOT 89
000090	001	90 FORMAT (27X,22HCONVENTIONAL TV SYSTEM/13H WATER TYPE: ,2A6,14X,4HSDAOT 90	
000091	001	SOURCE: ,2A6)	DAOT 91
000092	001	100 FORMAT (16H WAVE HEIGHT,FT:,F12.5,11X,14HSOURCE POWER:,F12.5)	DAOT 92
000093	001	110 FORMAT (11H H0ITUM,FT:,F12.5,16X,12HSOURCE REAM:,F12.5)	DAOT 93
000094	001	120 FORMAT (11H LENGTH,FT:,F12.5,16X,4HRECEIVER:,2A6)	DAOT 94
000095	001	130 FORMAT (11H HEIGHT,FT:,F12.5,16X,14HRCFIVER REAM:,F12.5)	DAOT 95
000096	001	140 FORMAT (5H TAU:,F12.5)	DAOT 96
000097	001	150 FORMAT (11H S.W. SEP.,F12.5,16X,3HFF:,F12.5)	DAOT 97
000098	001	160 FORMAT (110H RANGE,FT:,F12.5,17X,9HRANGE,AL:,F12.5)	DAOT 98
000099	001	170 FORMAT (125H AVAILARLE LINES AT CTN.,F12.5,2X,4HAT EDGE:,F12.5)	DAOT 99
000100	001	180 FORMAT (20H CONTRAST AT CENTER:,F12.5,7X,4HAT EDGE:,F12.5)	DAOT100
000101	001	190 FORMAT (126H AVAILARLE L H S SW WIDTH=:F9.5,13H FT ACTUAL =:F9.5,DAOT101	
000102	001	13H FT)	DAOT102
000103	001	200 FORMAT (12H LIMITED BY:,A6)	DAOT103
000104	001	210 FORMAT (126H AVAILARLE R H S SW WIDTH=:F9.5,13H FT ACTUAL =:F9.5,DAOT104	
000105	001	13H FT)	DAOT105
000106	001	220 FORMAT (126H AVAILARLE SW WIDTH=:F9.5,13H FT ACTUAL =:F9.5,3H FT)DAOT106	
000107	001	END	DAOT107-

BEST AVAILABLE COPY

000001	001	FUNCTION DFUNC(THET,SRSEP,RAN,EANG,IWAT)	DFUN 1
000002	001	C	DFUN 2
000003	001	C	DFUN 3
000004	001	COMMON BLOCK OF COMPUTING CONSTANTS	DFUN 4
000005	001	COMMON /IVLOW/ A(2,2), R(2,11,2), C(20),U(2,2), E(10),	DFUN 5
000006	001	1 F(2,2), G(2,2,3), H(2,2,3), N(10), O(15), P(2,2),	DFUN 6
000007	001	2 Q(15), S(2), T(11), W(10), Z(2), PI	DFUN 7
000008	001	C	DFUN 8
000009	001	U1=ATAN((SRSEP/RAN*TAN(THET-EANG))/RAN)-EANG	DFUN 9
000010	001	U2=((O(IWAT,1)/COS(O1+EANG))+(1/COS(THET-EANG)))	DFUN 10
000011	001	U3=C(1)*EXP(-O2*A(IWAT,1)*RAN)	DFUN 11
000012	001	U4=U3*(COS(EANG+O1)*3.)*(COS(THET)*4)/(RAN*RAN)	DFUN 12
000013	001	DFUNC=U4	DFUN 13
000014	001	RETURN	DFUN 14-
000001	001	FUNCTION EFUNC(CURR,J)	EFUN 1
000002	001	C	EFUN 2
000003	001	C	EFUN 3
000004	001	COMMON BLOCK OF COMPUTING CONSTANTS	EFUN 4
000005	001	COMMON /IVLOW/ A(2,2), R(2,11,2), C(20),U(2,2), E(10),	EFUN 5
000006	001	1 F(2,2), G(2,2,3), H(2,2,3), N(10), O(15), P(2,2),	EFUN 6
000007	001	2 Q(15), S(2), T(11), W(10), Z(2), PI	EFUN 7
000008	001	C	EFUN 8
000009	001	EFUNC=(ALOG((CURR*Z(J))/FM(J,1))/FM(J,2))	EFUN 9
000010	001	RETURN	EFUN 10-
000001	001	FUNCTION FNK (I,IB)	FNK 1
000002	001	C	FNK 2
000003	001	C	FNK 3
000004	001	COMMON BLOCK OF COMPUTING CONSTANTS	FNK 4
000005	001	COMMON /IVLOW/ A(2,2), R(2,11,2), C(20),U(2,2), E(10),	FNK 5
000006	001	1 F(2,2), G(2,2,3), H(2,2,3), N(10), O(15), P(2,2),	FNK 6
000007	001	2 Q(15), S(2), T(11), W(10), Z(2), PI	FNK 7
000008	001	C	FNK 8
000009	001	IND=1	FNK 9
000010	001	DO 10 IND=1,IB	FNK 10
000011	001	E(IND)=U2	FNK 11
000012	001	10 IND=IND+1	FNK 12
000013	001	IR=IR-1	FNK 13
000014	001	DO 20 IND=1,IB1	FNK 14
000015	001	IF (W(IND).GT.W(IND+1)) GO TO 20	FNK 15
000016	001	W(IND+1)=W(IND)	FNK 16
000017	001	E(IND+1)=E(IND)	FNK 17
000018	001	20 CONTINUE	FNK 18
000019	001	RETURN	FNK 19-
000001	001	FUNCTION GFUNC(A,IRAN,ANGLER,EANG,SRSEP)	GFUN 1
000002	001	C	GFUN 2
000003	001	C	GFUN 3
000004	001	C	GFUN 4
000005	001	U1=((ALOG(R/A))/(RAN*TAN(ANGLER+EANG)-(SRSEP/2.)))	GFUN 5
000006	001	GFUNC=-(ALOG(.07/A))/U1	GFUN 6
000007	001	RETURN	GFUN 7
000008	001	END	GFUN 8-
000001	001	SUBROUTINE HLIMIT (TFLAG)	HLIM 1
000002	001	C	HLIM 2
000003	001	C	HLIM 3
000004	001	C	HLIM 4
000005	001	C	HLIM 5
000006	001	COMMON BLOCK OF OPTIONS	HLIM 6
000007	001	COMMON /OPTS/ IRUN,IPAKAMICAL	HLIM 7
000008	001	C	HLIM 8
000009	001	COMMON BLOCK OF I/O VARIABLES	HLIM 9
000010	001	COMMON /IOLIST/ NVIRON, ISIDE, INTPE, WAVEHT, BTMRUF, TARGLN,	HLIM 10
000011	001	1 TARGHT, SRSEP, SPWNN, SUELF, RUELF, OPTAN, FNUM, HFIGHT,	HLIM 11
000012	001	2 RANGMN, RANGE, KANGAL, HANGER, KANGMY, SWWTHL, SWWTHR, SWWTHH,	HLIM 12
000013	001	3 SWWTH, AVSWL, AVSWR, AVSW, LINCEN, LINEUG, CTCEN, CTNEUG,	HLIM 13
000014	001	4 NUFXML, NUFWR, LIMFLG	HLIM 14
000015	001	C	HLIM 15
000016	001	C	HLIM 16
000017	001	COMMON BLOCK OF COMPUTING CONSTANTS	HLIM 17
000018	001	COMMON /IVLOW/ A(2,2), R(2,11,2), C(20),U(2,2), E(10),	HLIM 18
000019	001	1 F(2,2), G(2,2,3), H(2,2,3), N(10), O(15), P(2,2),	HLIM 19
000020	001	2 Q(15), S(2), T(11), W(10), Z(2), PI	HLIM 20
000021	001	C	HLIM 21
000022	001	REAL LIMEN,LINEUG	HLIM 22
000023	001	C	HLIM 23
000024	001	REQUIRED DEPTH OF FIELD, FEET	HLIM 24
000025	001	UFIFLD=.5*(WAVEHT+BTMRUF)+TARGHT	HLIM 25
000026	001	C	HLIM 26
000027	001	INITIAL ESTIMATE MINIMUM RANGE, FEET	
000028	001	H1=(.05/AINVIRON,1)*9.281	

000027	001	ANGLER=.00872665*RDFLF	HLIM 27
000028	001	ANGLER=.00872665*SDFLF	HLIM 28
000029	001	10 EX=ATAN(SRSEP/(2.*K1))	HLIM 29
000030	001	IF (AMAX1(1+X+ANGLER*EX+ANGLES).LT.(PI+.49)) GO TO 20	HLIM 30
000031	001	K1=1.05*K1	HLIM 31
000032	001	GO TO 10	HLIM 32
000033	001	C	HLIM 33
000034	001	20 T1=DFIELD*(SRSEP/(TAN(ANGLES)+TAN(ANGLER)))	HLIM 34
000035	001	30 EX=ATAN(SRSEP/(2.*T1))	HLIM 35
000036	001	T2=DFIELD*(SRSEP/(TAN(ANGLES+EX)+TAN(ANGLER+EX)))	HLIM 36
000037	001	IF (T2.GT.0.) GO TO 40	HLIM 37
000038	001	T2=0.	HLIM 38
000039	001	GO TO 50	HLIM 39
000040	001	C	HLIM 40
000041	001	40 IF (ABS(1)-T2).LT..001) GO TO 50	HLIM 41
000042	001	T1=T2	HLIM 42
000043	001	GO TO 30	HLIM 43
000044	001	C	HLIM 44
000045	001	C SET MINIMUM RANGE, FEET	HLIM 45
000046	001	50 RANGMN=AMAX1(T2*K1)	HLIM 46
000047	001	RANGMN=AMAX1(RANGMN*SRSEP)	HLIM 47
000048	001	C	HLIM 48
000049	001	C	HLIM 49
000050	001	T1=SDFLF*.5	HLIM 50
000051	001	T2=SDFLF*.5	HLIM 51
000052	001	D(NVIRON,1)=CFUNC(1,T1,NVIRON)	HLIM 52
000053	001	D(NVIRON,2)=CFUNC(2,T2,NVIRON)	HLIM 53
000054	001	C	HLIM 54
000055	001	IF (IPUN.EQ.1) GO TO 70	HLIM 55
000056	001	IF (IFLAG.NE.0) GO TO 70	HLIM 56
000057	001	PRINT 170, RANGMN	HLIM 57
000058	001	IF (IPARAM.NE.1) GO TO 60	HLIM 58
000059	001	PRINT 160	HLIM 59
000060	001	MEAN (5,170) HEIGHT	HLIM 60
000061	001	GO TO 70	HLIM 61
000062	001	60 PRINT 140	HLIM 62
000063	001	MEAN (5,150) ICAL	HLIM 63
000064	001	70 CONTINUE	HLIM 64
000065	001	IF (ICAL.EQ.4) GO TO 90	HLIM 65
000066	001	IF (ICAL-2) 100,70,80	HLIM 66
000067	001	C	HLIM 67
000068	001	C GET MAXIMUM SWATH WIDTH	HLIM 68
000069	001	80 CONTINUE	HLIM 69
000070	001	CAL TVFUNC (-1,.3)	HLIM 70
000071	001	RANGER=0(1)*3.281	HLIM 71
000072	001	SWATHR=0(12)*3.281	HLIM 72
000073	001	IF (IPARAM.EQ.1) GO TO 110	HLIM 73
000074	001	GO TO 120	HLIM 74
000075	001	C	HLIM 75
000076	001	C GET MAXIMUM RANGE	HLIM 76
000077	001	90 CONTINUE	HLIM 77
000078	001	CAL TVFUNC (-1,.2)	HLIM 78
000079	001	RANGMX=0(1)*3.281	HLIM 79
000080	001	IF (ICAL.EQ.4) GO TO 80	HLIM 80
000081	001	IF (IPARAM.EQ.1) GO TO 110	HLIM 81
000082	001	GO TO 120	HLIM 82
000083	001	C	HLIM 83
000084	001	100 IF (IRUN.EQ.1.OR.IPARAM.EQ.1) GO TO 110	HLIM 84
000085	001	PRINT 160	HLIM 85
000086	001	MEAN (5,170) HEIGHT	HLIM 86
000087	001	C	HLIM 87
000088	001	C	HLIM 88
000089	001	C CALCULATE SINGLE RANGE	HLIM 89
000090	001	110 CONTINUE	HLIM 90
000091	001	IF (HEIGHT.EQ.0.) GO TO 120	HLIM 91
000092	001	CAL TVFUNC (HEIGHT,1)	HLIM 92
000093	001	120 CONTINUE	HLIM 93
000094	001	RANGE=0(1)*3.281	HLIM 94
000095	001	RANGAL=0(1)*A(NVIRON,1)	HLIM 95
000096	001	AVSWL=RANGL*TAN(ANGLER-0(13))+0.5*SRSEP	HLIM 96
000097	001	SWATHL=0(1)*3.281	HLIM 97
000098	001	AVSWH=RANGH*TAN(ANGLER+0(13))-0.5*SRSEP	HLIM 98
000099	001	SWATHR=0(12)*3.281	HLIM 99
000100	001	AVSW=AVSWL+AVSWR	HLIM 100
000101	001	SWATH=0(12)*3.281	HLIM 101
000102	001	C	HLIM 102
000103	001	C DISPLAY LIMIT IN LIMIT FLAG.	HLIM 103
000104	001	C SHOW RIGHT AND LEFT-HAND SIDE WIDTH INDICES: AVERAGE	HLIM 104
000105	001	C LIMITS AT CENTER AND AT EDGE: CONTRAST AT CENTER AND AT EDGE.	HLIM 105
000106	001	LIMFLAG=13	HLIM 106
000107	001	NDEWL=0(9)	HLIM 107
000108	001	NDEWR=0(11)	HLIM 108
000109	001	LIMFEN=13	HLIM 109
000110	001	LIMFEN=14	HLIM 110
000111	001	CTHFCEN=16	HLIM 111
000112	001	CTHFCEN=17	HLIM 112

000113	001	C		HLIM113
000114	001		RETURN	HLIM114
000115	001	C		HLIM115
000116	001	C		HLIM116
000117	001	C		HLIM117
000118	001		130 FORMAT (17H MINIMUM RANGE IS F9.5+54 FEET)	HLIM118
000119	001		140 FORMAT (46H ONE RANGE(1);MAX. RANGE(2) OR MAX. WIDTH(3);?)	HLIM119
000120	001		150 FORMAT (11)	HLIM120
000121	001		160 FORMAT (11H RANGE,FT;?)	HLIM121
000122	001		170 FORMAT (G10.0)	HLIM122
000123	001		END	HLIM123-

***** MAIN *****

000001	001	C	INPUT FILES:	MAIN 1
000002	001	C	UNIT 5 (CARD READER) -	MAIN 2
000003	001	C	DATA OF VARIABLE INITIALIZATION AS FOLLOWS:	MAIN 3
000004	001	C	ENVIRONMENT, TARGET DIMENSIONS, PLATFORM CHARACTER-	MAIN 4
000005	001	C	ISTICS, SOURCE CHARACTERISTICS, AND RECEIVER	MAIN 5
000006	001	C	CHARACTERISTICS.	MAIN 6
000007	001	C	OUTPUT FILES:	MAIN 7
000008	001	C	UNIT 6 (PRINTER)	MAIN 8
000009	001	C	INTERMEDIATE AND SUMMARY INFORMATION	MAIN 9
000010	001	C		MAIN 10
000011	001	C	COMMON BLOCK OF COMPUTING CONSTANTS	MAIN 11
000012	001		COMMON /IVCOM/ A(2,2), R(2,1,2), C(20), U(2,2), E(10),	MAIN 12
000013	001		1 F(2,2), G(2,2,3), H(2,2,3), N(10), O(15), P(2,2),	MAIN 13
000014	001		2 Q(15), S(2), T(11), W(10), Z(2), PII	MAIN 14
000015	001	C		MAIN 15
000016	001	C	COMMON BLOCK OF OPTIONS	MAIN 16
000017	001		COMMON /UPIS/ IRUN, IPARAM, ICAI	MAIN 17
000018	001	C		MAIN 18
000019	001	C		MAIN 19
000020	001	C	ATTN. CONST.	MAIN 20
000021	001		DATA ((A(I),J),J=1,2),I=1,2) / .252, .273, .049, .049/ ,	MAIN 21
000022	001	C		MAIN 22
000023	001	C	EFFECTIVE ALFA ARRAY	MAIN 23
000024	001		DATA ((U(I),J,K),I=1,2),J=1,11),K=1,2) / 1., 1., .87, .91, .69, .69,	MAIN 24
000025	001		1 .74, .62, .72, .44, .63, .48, .55, .33, .51, .29, .49, .24, .44,	MAIN 25
000026	001		2 .20, .42, .17, .41, 1., 1., .87, .91, .68, .78, .59, .72, .45,	MAIN 26
000027	001		3 .53, .74, .55, .28, .51, .25, .49, .18, .44, .15, .42, .14, .41/	MAIN 27
000028	001	C		MAIN 28
000029	001	C	NUMBER OF LINES MATRIX	MAIN 29
000030	001		DATA ((F(I),J),J=1,2),I=1,2) / .000127, .00863, 9.51E-14, .0133/	MAIN 30
000031	001	C		MAIN 31
000032	001	C	SOURCE RECEIVER MATRIX	MAIN 32
000033	001		DATA ((U(I),J,K),I=1,2),J=1,2),K=1,3) / .00361, .00106, .002,	MAIN 33
000034	001		1 .00095, .0136, .00325, .00747, .00304, .00610, .00287, .00292,	MAIN 34
000035	001		2 .00346/	MAIN 35
000036	001	C		MAIN 36
000037	001	C	SOURCE RECEIVER MATRIX (BACKSCATTER)	MAIN 37
000038	001		DATA ((U(I),J,K),I=1,2),J=1,2),K=1,3) / .00435, .00281, .00277,	MAIN 38
000039	001		1 .00214, .0162, .0134, .00422, .00424, .0090, .00430, .00642,	MAIN 39
000040	001		2 .00574/	MAIN 40
000041	001	C		MAIN 41
000042	001	C	LAG MATRIX	MAIN 42
000043	001		DATA ((P(I),J),J=1,2),I=1,2) / 3.70, 105.0, 5.79E-5, 169.0/	MAIN 43
000044	001	C		MAIN 44
000045	001	C	SCATTERING CONSTANT	MAIN 45
000046	001		DATA S / .238, .030/	MAIN 46
000047	001	C		MAIN 47
000048	001	C	ANGLES	MAIN 48
000049	001		DATA T / 0., .05, .5, 1., 2.5, 5., 7.5, 10., 20., 30., 90./	MAIN 49
000050	001	C		MAIN 50
000051	001	C	RECEIVER FACTOR	MAIN 51
000052	001		DATA Z / 1070000.0, 1./	MAIN 52
000053	001	C		MAIN 53
000054	001	C	PI	MAIN 54
000055	001		DATA PII / 3.14159265/	MAIN 55
000056	001	C		MAIN 56
000057	001	C	INITIALIZE THE NUMBER OF SUMMARIES COUNTER AND SET THE	MAIN 57
000058	001	C	EXECUTION-BRANCH FLAG FOR READING THE FIRST CARD OF INPUT.	MAIN 58
000059	001	C	THIS CARD SPECIFIES THE TYPE OF PROCESSING, BATCH OR DEMAND	MAIN 59
000060	001	C	TERMINAL.	MAIN 60
000061	001		ISTART=-1	MAIN 61
000062	001		ISUM=0	MAIN 62
000063	001	C		MAIN 63
000064	001	C	READ DATA INPUT. IF NO DATA IS GIVEN, STOP EXECUTION.	MAIN 64
000065	001		IN CALL DATAIN (ISTART,ISUM)	MAIN 65
000066	001		IFLAG=0	MAIN 66

000067	001	IF (ISTART.NE.1) GO TO 30	MAIN 67
000068	001		MAIN 68
000069	001	C INCREMENT THE NUMBER OF SUMMARIES COUNTER. PERFORM	MAIN 69
000070	001	C CALCULATIONS AND OUTPUT RESULTS.	MAIN 70
000071	001	20 ISUM=ISUM+1	MAIN 71
000072	001	CALL HLIMIT (IFLAG)	MAIN 72
000073	001	CALL DATAI (ISUM)	MAIN 73
000074	001	C	MAIN 74
000075	001	C RESET THE EXECUTION-BRANCH FLAG TO STOP EXECUTION. IF	MAIN 75
000076	001	C IN PARAMETRIC MODE, UPDATE THE NECESSARY PARAMETER. (IF THE	MAIN 76
000077	001	C VALUE OF A PARAMETER HAS BEEN CHANGED, SUBROUTINE UPDATE WILL	MAIN 77
000078	001	C SET THE EXECUTION-BRANCH FLAG TO PERFORM NEW CALCULATIONS.)	MAIN 78
000079	001	C IF IN NON-PARAMETRIC MODE, OR IF NO CHANGE IN THE PARAMETERS	MAIN 79
000080	001	C IS MADE, CONTINUE READING INPUT.	MAIN 80
000081	001	C ISTART=0	MAIN 81
000082	001	C IF (IPARAM.EQ.1) CALL UPDATE (ISTART)	MAIN 82
000083	001	C IFLAG=ISTART	MAIN 83
000084	001	C IF (ISTART.EQ.1) GO TO 20	MAIN 84
000085	001	C GO TO 10	MAIN 85
000086	001	C	MAIN 86
000087	001	C IF SUMMARY INFORMATION IS GIVEN, OUTPUT THE SUMMARY	MAIN 87
000088	001	C TABLE. STOP EXECUTION.	MAIN 88
000089	001	30 IF (ISUM.NE.0) CALL SUMMARY (IRUN,ISUM)	MAIN 89
000090	001	C STOP	MAIN 90
000091	001	C	MAIN 91
000092	001	C ENU	MAIN 92-

000001	001	SUBROUTINE SUMMARY (IRUN,ISUM)	SUMA 1
000002	001	C	SUMA 2
000003	001	C SUBROUTINE SUMMARY OUTPUTS COMPLETE SUMMARY INFORMATION	SUMA 3
000004	001	C ON THE VISUAL SEARCH SYSTEM PROGRAM	SUMA 4
000005	001	C	SUMA 5
000006	001	C DIMENSION NVIRON(8),WAVEHT(8),RTMHIF(8),TARGLN(8),TARGHT(8),	SUMA 6
000007	001	C 1 SHSEP(8),ISTPE(8),SPWRH(8),SDELF(8),IRTPE(8),	SUMA 7
000008	001	C 2 RDEL(8),OPTHAN(8),FNUM(8),AVSWH(8),AVSW(8),RANGE(8),	SUMA 8
000009	001	C 3 RANGMN(8),RANGMX(8),RANGEB(8),SWWTH(8),SWWTHL(8),SWWTHR(8),	SUMA 9
000010	001	C 4 SWWTHK(8),RANGAL(8),LINLEN(8),LINEDG(8),CTRCEN(8),CTREDG(8),	SUMA 10
000011	001	C 5 AVSWL(8),IAL(8),IAH(8),IPART2(8)	SUMA 11
000012	001	C DIMENSION NAME(14)	SUMA 12
000013	001	C REAL LINCEN,LINEUG	SUMA 13
000014	001	C	SUMA 14
000015	001	C DATA NAME /6H COA,4HSTAL,6H ,4H)FEP,	SUMA 15
000016	001	C 1 6H INC,4HAND,6H)HAL, ,4H)DO,	SUMA 16
000017	001	C 2 6H)MRC, ,4H)VAP,6H VID,4H)CUN,	SUMA 17
000018	001	C 3 6H ,4H)SIT/	SUMA 18
000019	001	C	SUMA 19
000020	001	C RETURN 0	SUMA 20
000021	001	C	SUMA 21
000022	001	C ICK=4	SUMA 22
000023	001	C IF (IRUN.EQ.1) ICK=8	SUMA 23
000024	001	C KNTR=0	SUMA 24
000025	001	C DO 70 I=1,ISUM	SUMA 25
000026	001	C KNTR=KNTR+1	SUMA 26
000027	001	C READ (4) NVIRON(KNTR),WAVEHT(KNTR),RTMHIF(KNTR)	SUMA 27
000028	001	C READ (4) TARGLN(KNTR),TARGHT(KNTR)	SUMA 28
000029	001	C READ (4) SHSEP(KNTR)	SUMA 29
000030	001	C READ (4) ISTPE(KNTR),SPWRH(KNTR),SDELF(KNTR)	SUMA 30
000031	001	C READ (4) IRTPE(KNTR),RDEL(KNTR),OPTHAN(KNTR),FNUM(KNTR)	SUMA 31
000032	001	C READ (4) RANGE(KNTR),RANGAL(KNTR),RANGMN(KNTR),RANGMX(KNTR),RANGEB(SUMA 32	
000033	001	C 1)K)TH)	SUMA 33
000034	001	C READ (4) LINCEN(KNTR),LINEUG(KNTR),CTRCEN(KNTR),CTREDG(KNTR)	SUMA 34
000035	001	C READ (4) AVSWL(KNTR),SWWTHL(KNTR),IAL(KNTR)	SUMA 35
000036	001	C READ (4) AVSWR(KNTR),SWWTHK(KNTR),IAH(KNTR)	SUMA 36
000037	001	C READ (4) AVSW(KNTR),SWWTH(KNTR),SWWTHR(KNTR)	SUMA 37
000038	001	C	SUMA 38
000039	001	C IF (KNTR.FU.ICK.OR.I.EQ.ISUM) GO TO 10	SUMA 39
000040	001	C GO TO 10	SUMA 40
000041	001	C	SUMA 41
000042	001	C 10 CONTINUE	SUMA 42
000043	001	C IF (IRUN.EQ.1) GO TO 20	SUMA 43
000044	001	C PRINT 00	SUMA 44
000045	001	C READ (5,90)	SUMA 45
000046	001	C GO TO 30	SUMA 46
000047	001	C 20 PRINT 100	SUMA 47
000048	001	C 30 CONTINUE	SUMA 48
000049	001	C	SUMA 49
000050	001	C DO 40 K=1,KNTR	SUMA 50
000051	001	C IN=NVIRON(K)+NVIRON(K)-1	SUMA 51
000052	001	C INP1=IN+1	SUMA 52
000053	001	C NVIRON(K)=NAME(IN)	SUMA 53
000054	001	C IPART2(K)=NAME(INP1)	SUMA 54
000055	001	C 40 CONTINUE	SUMA 55

000056	001	PRINT 110, (INVIION(K),IPART2(K)),K=1,KNTN)	SIMA 56
000057	001	PRINT 120, (WAVEHT(K),K=1,KNTN)	SIMA 57
000058	001	PRINT 130, (BTMRUF(K),K=1,KNTN)	SIMA 58
000059	001	PRINT 140, (TARGLN(K),K=1,KNTN)	SIMA 59
000060	001	PRINT 150, (TARGHT(K),K=1,KNTN)	SIMA 60
000061	001	PRINT 160, (SRSEF(K),K=1,KNTN)	SIMA 61
000062	001	DO 50 K=1,KNTN	SIMA 62
000063	001	IN=ISTH(K)+ISTPE(K)+3	SIMA 63
000064	001	INP1=IN+1	SIMA 64
000065	001	ISTPE(K)=NAME(IN)	SIMA 65
000066	001	IPART2(K)=NAME(INP1)	SIMA 66
000067	001	50 CONTINUE	SIMA 67
000068	001	PRINT 170, ((ISTPE(K),IPART2(K)),K=1,KNTN)	SIMA 68
000069	001	PRINT 180, (SPWM(K),K=1,KNTN)	SIMA 69
000070	001	PRINT 190, (SDELF(K),K=1,KNTN)	SIMA 70
000071	001	DO 60 K=1,KNTN	SIMA 71
000072	001	IN=INTH(K)+IPTPE(K)+9	SIMA 72
000073	001	INP1=IN+1	SIMA 73
000074	001	INTPE(K)=NAME(IN)	SIMA 74
000075	001	IPART2(K)=NAME(INP1)	SIMA 75
000076	001	60 CONTINUE	SIMA 76
000077	001	PRINT 200, ((INTPE(K),IPART2(K)),K=1,KNTN)	SIMA 77
000078	001	PRINT 210, (HDELF(K),K=1,KNTN)	SIMA 78
000079	001	PRINT 220, (OPTIRAN(K),K=1,KNTN)	SIMA 79
000080	001	PRINT 230, (FNUM(K),K=1,KNTN)	SIMA 80
000081	001	PRINT 240, (HANGE(K),K=1,KNTN)	SIMA 81
000082	001	PRINT 250, (HANGAL(K),K=1,KNTN)	SIMA 82
000083	001	PRINT 260, (HANGM(K),K=1,KNTN)	SIMA 83
000084	001	PRINT 270, (HANGMX(K),K=1,KNTN)	SIMA 84
000085	001	PRINT 280, (HANGE(K),K=1,KNTN)	SIMA 85
000086	001	PRINT 290, (LINC(K),K=1,KNTN)	SIMA 86
000087	001	PRINT 300, (LINF(K),K=1,KNTN)	SIMA 87
000088	001	PRINT 310, (CTHCLN(K),K=1,KNTN)	SIMA 88
000089	001	PRINT 320, (CTHFLG(K),K=1,KNTN)	SIMA 89
000090	001	PRINT 330, (AVSWL(K),K=1,KNTN)	SIMA 90
000091	001	PRINT 340, (SWWTL(K),K=1,KNTN)	SIMA 91
000092	001	PRINT 350, (IAL(K),K=1,KNTN)	SIMA 92
000093	001	PRINT 360, (AVSWH(K),K=1,KNTN)	SIMA 93
000094	001	PRINT 370, (SWWTH(K),K=1,KNTN)	SIMA 94
000095	001	PRINT 380, (IAH(K),K=1,KNTN)	SIMA 95
000096	001	PRINT 390, (AVSW(K),K=1,KNTN)	SIMA 96
000097	001	PRINT 400, (SWWTH(K),K=1,KNTN)	SIMA 97
000098	001	PRINT 410, (SWWTHH(K),K=1,KNTN)	SIMA 98
000099	001	C	SIMA 99
000100	001	KNTH=0	SIMA100
000101	001	70 CONTINUE	SIMA101
000102	001	ISUM=0	SIMA102
000103	001	C	SIMA103
000104	001	NEFINN	SIMA104
000105	001	C	SIMA105
000106	001	C	SIMA106
000107	001	C	SIMA107
000108	001	80 FORMAT (24H CLEAR SCREEN AND RETURN)	SIMA108
000109	001	90 FORMAT (80A1)	SIMA109
000110	001	100 FORMAT (1H1)	SIMA110
000111	001	110 FORMAT (12H WATER TYPE:11X,8(2X,A6,A4))	SIMA111
000112	001	120 FORMAT (16H WAVE HEIGHT:FT:7X,AF12.4)	SIMA112
000113	001	130 FORMAT (21H BOTTOM ROUGHNESS:FT:2X,8F12.4)	SIMA113
000114	001	140 FORMAT (18H TARGET LENGTH:FT:5X,AF12.4)	SIMA114
000115	001	150 FORMAT (18H TARGET HEIGHT:FT:5X,AF12.4)	SIMA115
000116	001	160 FORMAT (13H S-R-SEP:FT:10X,AF12.4)	SIMA116
000117	001	170 FORMAT (8H SOURCE:15X,8(2X,A6,A4))	SIMA117
000118	001	180 FORMAT (20H SOURCE POWER,WATTS:3X,AF12.4)	SIMA118
000119	001	190 FORMAT (17H SOURCE REAM:UF:6X,8F12.4)	SIMA119
000120	001	200 FORMAT (10H RECEIVER:13X,8(2X,A6,A4))	SIMA120
000121	001	210 FORMAT (19H RECEIVER REAM:UF:4X,8F12.4)	SIMA121
000122	001	220 FORMAT (5H TAU:14X,8F12.4)	SIMA122
000123	001	230 FORMAT (10H F-NUMBER:13X,8F12.4)	SIMA123
000124	001	240 FORMAT (10H RANGE:FT:13X,8F12.4)	SIMA124
000125	001	250 FORMAT (2X,10H RANGE:AL:11X,AF12.4)	SIMA125
000126	001	260 FORMAT (2X,12H MINIMUM:FT:9X,8F12.4)	SIMA126
000127	001	270 FORMAT (2X,12H MAXIMUM:FT:9X,8F12.4)	SIMA127
000128	001	280 FORMAT (2X,9H BEST:FT:12X,AF12.4)	SIMA128
000129	001	290 FORMAT (17H AV LINES AT CTK:6X,8F12.4)	SIMA129
000130	001	300 FORMAT (2X,9H AT FURE:12X,AF12.4)	SIMA130
000131	001	310 FORMAT (17H CONTRAST AT CTK:6X,8F12.4)	SIMA131
000132	001	320 FORMAT (23H AV L-H-S. SW WIDTH:FT:AF12.4)	SIMA132
000133	001	330 FORMAT (2X,11H ACTUAL:FT:10X,8F12.4)	SIMA133
000134	001	340 FORMAT (2X,14H - LIMITED AT:7X,8(6X,A6))	SIMA134
000135	001	350 FORMAT (23H AV R-H-S. SW WIDTH:FT:AF12.4)	SIMA135
000136	001	360 FORMAT (19H AV SWATH WIDTH:FT:4X,8F12.4)	SIMA136
000137	001	END	SIMA137-

000001	001	SUBROUTINE TVFUNC (RSET,ICAL)	TVFU 1
000002	001	C	TVFU 2
000003	001	C	TVFU 3
000004	001	C	TVFU 4
000005	001	COMMON /IOLIST/ NVIRON, ISIZE, IRTPE, WAVEHT, BTHRU, TARGLN,	TVFU 5
000006	001	1 TARGHT, SRSEP, SPOWR, SUELF, RUELF, OTRAN, FNUM, HEIGHT,	TVFU 6
000007	001	2 RANGMN, RANGE, RANGAL, RANGER, RANGMX, SWTHL, SWTHR, SWTHR,	TVFU 7
000008	001	3 SWTH, AVSWL, AVSWR, AVSW, LINCEN, LINEUG, CTRCN, CTNEDG,	TVFU 8
000009	001	4 NUFXWL, NUFXWH, LIMFLG	TVFU 9
000010	001	C	TVFU 10
000011	001	C	TVFU 11
000012	001	COMMON /IVCOM/ A(2,2), R(2,11,2), C(20), D(2,2), E(10),	TVFU 12
000013	001	1 F(2,2), G(2,2,3), H(2,2,3), N(10), O(15), P(2,2),	TVFU 13
000014	001	2 Q(15), S(2), T(11), W(10), Z(2), PII	TVFU 14
000015	001	C	TVFU 15
000016	001	HEAL L1 NCEN,LINEUG	TVFU 16
000017	001	C	TVFU 17
000018	001	I=NVIRON	TVFU 18
000019	001	J=IRTP	TVFU 19
000020	001	K=JCTPE	TVFU 20
000021	001	C	TVFU 21
000022	001	C	TVFU 22
000023	001	C	TVFU 23
000024	001	COMPUTE IN METERS: DEPTH OF FIELD, SOURCE-RECEIVER	TVFU 24
000025	001	SEPARATION, AND VELOCITY(METERS/SEC).	TVFU 25
000026	001	UF=.3048*(.5*(WAVEHT+RTMHT)+TARGHT)	TVFU 26
000027	001	ANGLER=.00872665*RUELF	TVFU 27
000028	001	ANGLES=.00872665*SUELF	TVFU 28
000029	001	SRSEP1=.3048*SRSEP	TVFU 29
000030	001	SRSEP2=.5*SRSEP1	TVFU 30
000031	001	C	TVFU 31
000032	001	C	TVFU 32
000033	001	COMPUTE SIGNAL CONSTANT AND BACKSCATTER CONSTANT	TVFU 33
000034	001	C(1)=.00872665*SPOWR*OPHTAN*(1,0,0)	TVFU 34
000035	001	C(1)=C(1)/(128.*PI*(1.0-COS(ANGLES))*FNUM*FNUM)	TVFU 35
000036	001	C(2)=C(1)*S(1)*.02*H(1,0,0)/(1.5*G(1,0,0))	TVFU 36
000037	001	C	TVFU 37
000038	001	C	TVFU 38
000039	001	C	TVFU 39
000040	001	INITIALIZE N, Q, AND O	TVFU 40
000041	001	DO 10 I=1,10	TVFU 41
000042	001	N(I)=0	TVFU 42
000043	001	Q(I)=0	TVFU 43
000044	001	O(I)=0	TVFU 44
000045	001	10 CONTINUE	TVFU 45
000046	001	DO 20 I=11,15	TVFU 46
000047	001	N(I)=0	TVFU 47
000048	001	Q(I)=0	TVFU 48
000049	001	O(I)=0	TVFU 49
000050	001	20 CONTINUE	TVFU 50
000051	001	C	TVFU 51
000052	001	C	TVFU 52
000053	001	SET OR CHOOSE RANGE IN METERS	TVFU 53
000054	001	RANGE=.3048*SRSEP	TVFU 54
000055	001	IF (ICAL.EQ.1) GO TO 30	TVFU 55
000056	001	RANGE=.3048*RANGMN	TVFU 56
000057	001	RANGE=RANGI	TVFU 57
000058	001	30 C(4)=1./A(1,1)	TVFU 58
000059	001	40 U(1)=RANG	TVFU 59
000060	001	EANG=ATAN(SRSEP1/(RANG+RANG))	TVFU 60
000061	001	DELTA1=TAN(ANGLER+EANG)	TVFU 61
000062	001	DELTA2=TAN(ANGLER+EANG)	TVFU 62
000063	001	C(5)=RANG*(DELTA1+DELTA2)	TVFU 63
000064	001	C(5)=C(5)/(1.3048*TARGLN)	TVFU 64
000065	001	IF (C(5).LT.525.) GO TO 50	TVFU 65
000066	001	N(3)=1	TVFU 66
000067	001	U(1)=0.	TVFU 67
000068	001	GO TO 150	TVFU 68
000069	001	50 N(4)=1	TVFU 69
000070	001	N(3)=0	TVFU 70
000071	001	T1=0.	TVFU 71
000072	001	T2=0.	TVFU 72
000073	001	C	TVFU 73
000074	001	C	TVFU 74
000075	001	COMPUTE: SIGNAL CURRENT AT CENTER, OBJECT LINES AT	TVFU 75
000076	001	CENTER, AND OBJECT LINES AT EDGE	TVFU 76
000077	001	C(6)=PI*INC(0,SRSEP1,RANG,EANG,NVIRON)	TVFU 77
000078	001	T5=FINC(C(6),IRTP)	TVFU 78
000079	001	O(3)=AMIN1((T5*A.)/C(5),4200./C(5))	TVFU 79
000080	001	T5=ANGLER	TVFU 80
000081	001	C(7)=PI*INC(T5,SRSEP1,RANG,EANG,NVIRON)	TVFU 81
000082	001	T5=FINC(C(7),IRTP)	TVFU 82
000083	001	O(4)=AMIN1((T5*A.)/C(5),4200./C(5))	TVFU 83
000084	001	C	TVFU 84
000085	001	C	TVFU 85
000086	001	IF (T5.A.L(5)) GO TO 110	TVFU 86
000087	001	T1=1.	TVFU 87
000088	001	C	TVFU 88
000089	001	C	TVFU 89
000090	001	COMPUTE SIGNAL CURRENT AND NUMBER OF LINES	TVFU 90

000086	001	T5=ANGLE	TVF0 A6
000087	001	60 T6=DFINC(T5,SRSEP1,RANG,EANG,NVIRON)	TVF0 A7
000088	001	T6=DFINC(T6,IRIPE)	TVF0 A8
000089	001	IF (T6,GI,C(5)) GO TO 70	TVF0 A9
000090	001	T3=1.	TVF0 90
000091	001	GO TO 80	TVF0 91
000092	001	C	TVF0 92
000093	001	SIGNAL LIMITED?	TVF0 93
000094	001	70 T3=0.	TVF0 94
000095	001	80 IF (T3,LI,.5) GO TO 100	TVF0 95
000096	001	T1=T3	TVF0 96
000097	001	T5=T5+.05*(ANGLEH+EANG)	TVF0 97
000098	001	IF (ARS(T5-FANG).LT.(.5+EANG)) GO TO 90	TVF0 98
000099	001	GO TO 60	TVF0 99
000100	001	C	TVF0100
000101	001	ZERO SWATH WIDTH DUE TO POWER LIMIT	TVF0101
000102	001	90 U(5)=1.	TVF0102
000103	001	W(1)=SRSEP2	TVF0103
000104	001	W(4)=SRSEP2	TVF0104
000105	001	U(2)=0.	TVF0105
000106	001	GO TO 120	TVF0106
000107	001	C	TVF0107
000108	001	NORMAL EXIT FROM POWER LIMIT	TVF0108
000109	001	100 U(5)=1.	TVF0109
000110	001	U(2)=T5-(.05*(ANGLEH+EANG))	TVF0110
000111	001	T4=RANG+TAN(EANG-T5)	TVF0111
000112	001	W(1)=T4+SRSEP2	TVF0112
000113	001	W(4)=T4-SRSEP2	TVF0113
000114	001	GO TO 120	TVF0114
000115	001	C	TVF0115
000116	001	MAXIMUM WIDTH	TVF0116
000117	001	110 W(1)=RANG*(DELTA1+SRSEP2	TVF0117
000118	001	W(4)=RANG*(DELTA2-SRSEP2	TVF0118
000119	001	U(5)=1.	TVF0119
000120	001	C	TVF0120
000121	001	120 T1=VFINC(0.,SRSEP1,FANG,RANG,ANGLES,NVIRON)	TVF0121
000122	001	U(6)=.5*(C(6)/C(6)+T1)	TVF0122
000123	001	T1=VFINC(1-ANGLEH,SRSEP1,EANG,RANG,ANGLES,NVIRON)	TVF0123
000124	001	U(7)=.5*(C(7)/C(7)+T1)	TVF0124
000125	001	IF (U(7).LT..07) GO TO 130	TVF0125
000126	001	W(1)=RANG*(DELTA2-SRSEP2	TVF0126
000127	001	GO TO 140	TVF0127
000128	001	C	TVF0128
000129	001	130 W(1)=DFINC(0(6),U(7),RANG,ANGLEH,EANG,SRSEP1)	TVF0129
000130	001	140 W(2)=VFINC(DELTA1,RANG,DF,SRSEP1)	TVF0130
000131	001	T4=TAN(ANGLES+EANG)	TVF0131
000132	001	W(3)=DFINC(T4,RANG,DF,SRSEP1)	TVF0132
000133	001	W(5)=DFINC(DELTA2,RANG,DF,SRSEP1)	TVF0133
000134	001	T4=TAN(ANGLES-EANG)	TVF0134
000135	001	W(6)=VFINC(T4,RANG,DF,SRSEP1)	TVF0135
000136	001	CALL FNK (1,3)	TVF0136
000137	001	O(4)=W(1)	TVF0137
000138	001	U(1)=F(1)	TVF0138
000139	001	CALL FNK (4,7)	TVF0139
000140	001	U(10)=W(1)	TVF0140
000141	001	O(11)=F(1)	TVF0141
000142	001	U(12)=O(8)+O(10)	TVF0142
000143	001	IF (O(12).LT.0.) O(12)=0.	TVF0143
000144	001	U(13)=EANG	TVF0144
000145	001	C	TVF0145
000146	001	150 CONTINUE	TVF0146
000147	001	IF (IFLAG,0.1) GO TO 160	TVF0147
000148	001	IF (ICAL,0.1) GO TO 230	TVF0148
000149	001	IF (O(12).LE.0.) GO TO 230	TVF0149
000150	001	IFLAG=1	TVF0150
000151	001	GO TO 230	TVF0151
000152	001	C	TVF0152
000153	001	160 IF (ARS(C(4)).GE..05/A(1,1)) GO TO 170	TVF0153
000154	001	IFLAG=0	TVF0154
000155	001	GO TO 230	TVF0155
000156	001	C	TVF0156
000157	001	170 IF (ICAL=2) 230,140,200	TVF0157
000158	001	180 IF (O(12).GT.0.) GO TO 190	TVF0158
000159	001	IFLAG=1	TVF0159
000160	001	C(4)=-.5*ARS(C(4))	TVF0160
000161	001	GO TO 230	TVF0161
000162	001	C	TVF0162
000163	001	190 IF (IFLAG2,0.0) GO TO 230	TVF0163
000164	001	C(4)=.5*ARS(C(4))	TVF0164
000165	001	GO TO 230	TVF0165
000166	001	C	TVF0166
000167	001	200 IF (O(12).GT.O(12)) GO TO 230	TVF0167
000168	001	IF (O(12).GT.0.) GO TO 220	TVF0168
000169	001	IF (RANG=1.5*ARS(C(4)).GT.RANG1) GO TO 210	TVF0169

000170	001	RANG=RANG-ARS(C(4))	TVFU170
000171	001	C(4)=.5*ARS(C(4))	TVFU171
000172	001	GO TO 230	TVFU172
000173	001	C	TVFU173
000174	001	210 RANG=RANG-2.*ABS(C(4))	TVFU174
000175	001	C(4)=.5*ARS(C(4))	TVFU175
000176	001	GO TO 230	TVFU176
000177	001	C	TVFU177
000178	001	220 C(4)=-.5*C(4)	TVFU178
000179	001	230 DO 240 I=1,13	TVFU179
000180	001	U(I)=U(I)	TVFU180
000181	001	240 CONTINUE	TVFU181
000182	001	IF (IFLAG.EQ.0) GO TO 250	TVFU182
000183	001	RANG=RANG+C(4)	TVFU183
000184	001	GO TO 40	TVFU184
000185	001	C	TVFU185
000186	001	250 RETURN	TVFU186
000187	001	END	TVFU187-

000001	001	FUNCTION U=UNC(A,RAN,DFIELD,SRSEP)	UFUN 1
000002	001	UFUNC=A*(RAN-(SRSEP/(2.*A))-DFIELD)	UFUN 2
000003	001	RETURN	UFUN 3
000004	001	END	UFUN 4-

000001	001	SHORTLINE UPDATE (ISTART)	UPDA 1
000002	001	C	UPDA 2
000003	001	C	UPDA 3
000004	001	C	UPDA 4
000005	001	COMMON BLOCK OF I/O VARIABLES	UPDA 5
000006	001	COMMON /IOLIST/ NVIRON, ISTPE, INTPE, WAVEHT, BTMRUF, TARGLN,	UPDA 6
000007	001	1 TADGHI, SRSEP, SPOWR, SUFLF, RUFLF, OPTAN, FNUM, HFIGHT,	UPDA 7
000008	001	2 RANGMI, RANG, RANGAL, RANGER, RANGMX, SWHTML, SWHTHR, SWHTMR,	UPDA 8
000009	001	3 SWHTI, AVSWL, AVSWR, AVSW, LINCEN, LINEUG, CTCEN, CTKEOG,	UPDA 9
000010	001	4 NUFAWL, NUEXWK, LIMFLG	UPDA 10
000011	001	C	UPDA 11
000012	001	COMMON BLOCK OF UPDATE INFORMATION	UPDA 12
000013	001	COMMON /UPI/ STEP(14), EDVAL(14)	UPDA 13
000014	001	C	UPDA 14
000015	001	DIMENSION VALUE(14)	UPDA 15
000016	001	DIMENSION NEG(14),ID(14)	UPDA 16
000017	001	DATA NTRY /0/	UPDA 17
000018	001	C	UPDA 18
000019	001	EQUIVALENCE (VALUE(4),WAVEHT)	UPDA 19
000020	001	C	UPDA 20
000021	001	VALUE(1)=NVIRON	UPDA 21
000022	001	VALUE(2)=ISTPE	UPDA 22
000023	001	VALUE(3)=INTPE	UPDA 23
000024	001	IF (NTRY.NE.0) GO TO 20	UPDA 24
000025	001	NTRY=0	UPDA 25
000026	001	DO 10 I=1,14	UPDA 26
000027	001	IF (ARS(STEP(I)).EQ.0.) GO TO 10	UPDA 27
000028	001	NTRY=NTRY+1	UPDA 28
000029	001	ID(NTRY)=1	UPDA 29
000030	001	BEQ(NTRY)=VALUE(I)	UPDA 30
000031	001	10 CONTINUE	UPDA 31
000032	001	NTRY=1	UPDA 32
000033	001	C	UPDA 33
000034	001	20 IF (NTRY.EQ.0) GO TO 70	UPDA 34
000035	001	I=1	UPDA 35
000036	001	30 IN=I(1)	UPDA 36
000037	001	IF (VALUE(IN).NE.EDVAL(IN)) GO TO 50	UPDA 37
000038	001	IF (I.NE.NTRY) GO TO 40	UPDA 38
000039	001	NTRY=N	UPDA 39
000040	001	GO TO 70	UPDA 40
000041	001	40 VALUE(IN)=HF6(I)	UPDA 41
000042	001	I=I+1	UPDA 42
000043	001	GO TO 30	UPDA 43
000044	001	C	UPDA 44
000045	001	50 ISTART=1	UPDA 45
000046	001	VALUE(IN)=VALUE(IN)+STEP(IN)	UPDA 46
000047	001	IF (ARS(STEP(IN)).EQ.STEP(IN)) GO TO 60	UPDA 47
000048	001	IF (VALUE(IN).LT.EDVAL(IN)) VALUE(IN)=EDVAL(IN)	UPDA 48
000049	001	GO TO 70	UPDA 49
000050	001	60 IF (VALUE(IN).GT.EDVAL(IN)) VALUE(IN)=EDVAL(IN)	UPDA 50
000051	001	C	UPDA 51
000052	001	70 NVIRON=VALUE(1)	UPDA 52
000053	001	ISTPE=VALUE(2)	UPDA 53
000054	001	INTPE=VALUE(3)	UPDA 54
000055	001	C	UPDA 55
000056	001	RETURN	UPDA 56
000057	001	C	UPDA 57-
		END	


```

000001      UN1      FUNCTION VFUNC(A,RAN,DFIELD,SRSEP)
000002      UN1      C          CALCULATE SOURCE H. AND RECEIVED L WIDTH
000003      UN1      VFUNC=A*(RAN*(SRSEP/(2.*A))-DFIELD)
000004      UN1      RETURN
000005      UN1      ENJ

```

```

VFUN  1
VFUN  2
VFUN  3
VFUN  4
VFUN  5-

```

```

000001      UN1      FUNCTION XFUNC(THET,SRSEP,EANG,RAN,ANG,T)
000002      UN1      C
000003      UN1      C          COMMON BLOCK OF COMPUTING CONSTANTS
000004      UN1      COMMON /IVCOM/ A(2,2), R(2,11,2), C(20),U(2,2), E(10),
000005      UN1      1 F4(2,2), G(2,2,3), H(2,2,3), N(10), O(15), P(2,2),
000006      UN1      2 Q(15), S(2), T(11), W(10), Z(2), PI
000007      UN1      C
000008      UN1      F1=(SRSEP/(TAN(EANG+ANG)+TAN(EANG-THET)))
000009      UN1      F2=COS(THET-EANG)
000010      UN1      F3=((N(1,1)*A(1,1))+((N(1,2)*A(1,2)))/F2
000011      UN1      F4=((A*FUNC(F3*F1))/F1-(A*FUNC(F3*HANI)/RAN))
000012      UN1      XFUNC=L(2)*COS(THET)*F4)+2*F4
000013      UN1      RETURN
000014      UN1      ENJ

```

```

XFUN  1
XFUN  2
XFUN  3
XFUN  4
XFUN  5
XFUN  6
XFUN  7
XFUN  8
XFUN  9
XFUN 10
XFUN 11
XFUN 12
XFUN 13
XFUN 14-

```

BEST AVAILABLE COPY